

**Jihočeská univerzita v Českých Budějovicích**

Pedagogická fakulta – Katedra fyziky

**Návrh a realizace přístroje na sledování a vyhodnocování pohybové  
aktivity hmyzu**

Diplomová práce

Vedoucí práce: Ing. Michal Šerý

Autor: Bc. Štefan Kakaš

## **Anotace**

Diplomová práce se zabývá praktickým řešením konstrukce hardware a vývojem software pro přístroj určený ke sledování a vyhodnocování pohybové aktivity hmyzu. Úvodní část obsahuje rešerši vlastností a schopností dostupných software z této oblasti na trhu. V následujících kapitolách je následně řešena konstrukční část hardware přístroje a charakter speciálně vyvinutého software DROSANA. V závěru jsou porovnány analyzovaná data přístroje s konkurenčním řešením obdobných zařízení.

## **Anotation**

This diploma thesis deals with the practical solution design of hardware and software development for the device, which can monitoring and evaluating physical activity of insects. The introductory part includes search features and capabilities of the software available from this area in the Marketplace. In the following chapters is subsequently dealt with component hardware devices and character specially developer software DROSANA. Finally, the conclusion of thesis consists comparing measured data with developed devices and others competitive facilities.

## **Abstrakt**

Cílem diplomové práce je vývoj a následná konstrukce přístroje, jehož funkcí bude měření a vyhodnocování pohybové aktivity hmyzu.

Tento přístroj, v ČR ojedinělí, bude schopen záznamu pohybové aktivity hmyzu pomocí kamery ve formě videa. Následně bude schopen z takto natočeného videa pomocí software naprogramovaného ve Visual Basic .NET zpětně rekonstruovat kinematická a statistická data o pohybu hmyzu. Na základě dostatečného množství takto získaných dat mohou být ověřovány premisy biologů o funkci nervové soustavy, funkci mozku a behaviorálního chování Octomilky Obecné (*Drosophila Melangostar*) jež je více než jedno století používána jako genetický model pro experimenty.

Závěr práce porovnává výsledky doposud známých a dostupných softwarových řešení v této oblasti biologie, které jsou doposud používány jako zdroj výsledků pro prováděné experimenty.

**Klíčová slova** – video, obrazová analýza, *Drosophila Melangostar*, genetika; VB.NET

## **Abstract**

The aim of the diploma thesis is the development and subsequent construction of the device whose function is measuring and evaluating physical activity of insects.

This instrument is unique in the Czech Republic and will be able to record physical activity of insects by using a video camera. Later that it will be able in this video shot back to reconstruct kinematic and statistical data about the movement of insects with software which is developed with Visual Basic .NET. On the base sufficient od obtained data it can be used to verify the premise of biologists about nervous systém, brain function and behavioral of *Drosophila Melangostar* which is more than a century used as model for genetics experiments.

Finally, the conclusion compares the result of previously known and available software solutions in the world of biology, which are still used as a source for the results of performed experiments.

**Key words** – video, image analysis, *Drosophila Melangostar*, genetic; VB.NET

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně, pouze s použitím pramenů literatury, uvedených v seznamu literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své diplomové práce, a to v nezkrácené podobě archivované Pedagogickou fakultou elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 5.1.2012

Podpis studenta

Touto formou děkuji svému konzultantovi Ing. Michalu Šerému, za cenné rady a připomínky při zpracování mé práce, dále všem pracovníkům Entomologického ústavu Biologického centra AV ČR v Českých Budějovicích.

## OBSAH:

1 ÚVOD.....	8
2 METODY A SOFTWARE PRO MĚŘENÍ POHYBOVÉ AKTIVITY HMYZU .	9
2.1 TriKinetics .....	9
2.1.1 Popis systému .....	9
2.1.2 Popis jednotlivých částí .....	10
2.1.3 Schopnosti HW a nároky SW .....	11
2.1.4 Pořizovací náklady.....	11
2.1.5 Klady a zápory .....	12
2.2 PySolo.....	12
2.2.1 Popis systému .....	12
2.2.2 Schopnosti HW a nároky SW .....	13
2.2.3 Pořizovací náklady.....	14
2.2.4 Klady a zápory .....	14
2.3 CTRAX.....	14
2.3.1 Popis systému .....	14
2.3.2 Schopnosti HW a nároky SW .....	15
2.3.3 Pořizovací náklady.....	16
2.3.4 Klady a zápory .....	16
2.4 Parametry plánované konstrukce přístroje.....	17
3 HW ŘEŠENÍ PŘÍSTROJE .....	19
3.1 Požadavky na konstrukci .....	19
3.2 Návrh a model konstrukce .....	19
3.2.1 Konstrukční tyče .....	19
3.2.2 Podsvícení konstrukce .....	20
3.2.3 Plato pro misky a trubičky .....	22
3.2.4 Kryt konstrukce.....	25
3.2.4 První zhotovení modelu nosné konstrukce .....	25
3.2.5 Finální konstrukce.....	28
3.3 Volba záznamového zařízení .....	29
3.3.1 Detailní hlediska výběru .....	30
3.3.2 Hardwarová úprava webové kamery .....	37
4 SW ŘEŠENÍ .....	46
4.1 Volba vývojového prostředí.....	46
4.2 Výběr názvu vyvíjeného software .....	47
4.3 Návrh grafického rozhraní software .....	47
4.4 Popis formulářů programu DROSANA.....	49
4.4.1 Úprava obrazu.....	49
4.4.2 Výběr regionů .....	50
4.4.3 Nastavení masek .....	51
4.4.4 Dodatečné filtry .....	53
4.4.5 Nastavení prahování .....	54
4.4.6 Nastavení měřítka .....	56
4.4.7 Nastavení trasování.....	57
4.4.8 Nastavení detekce .....	58

4.4.9 Nastavení výstupu.....	59
4.4.10 Nastavení externího podnětu .....	60
4.5 Návrh zdrojového kódu .....	61
4.6 Popis částí zdrojového kódu .....	63
<b>5 POROVNÁNÍ NAMĚŘENÝCH VÝSLEDKŮ .....</b>	<b>80</b>
<b>6 ZÁVĚR .....</b>	<b>87</b>
<b>7 PŘÍLOHY .....</b>	<b>88</b>

## 1 ÚVOD

Výzkum v oblasti pohybové aktivity a samotného chování hmyzu, zejména pak ovocných mušek (octomilek), které se rády shromažďují kolem zkaženého ovoce, se může z pohledu laika zdát nepodstatné až úsměvné, avšak pro biology a vědce na celém světě je to již po zhruba jedno století cenný zdroj informací, který může pomoci při léčbě mnoha genetických onemocnění.

Od roku 1933 kdy Thomas Hunt Morgan získal Nobelovu cenu za demonstraci premisy, která tvrdí, že genetická informace je uchovávána v chromozomech buňky, kde mu jako zvířecí model posloužila právě *Drosophila Melangostar* (Octomilka obecná) se tak tato muška od té doby stala nejužívanějším genetickým modelem, díky svému krátkému měsíčnímu životnímu cyklu, snadnému chovu v laboratorních podmínkách a v neposlední řadě i díky snadné genové manipulaci. Světový výzkum v této oblasti podpořený Nobelovou cenou za medicínu v roce 1995 (genetická kontrola raného embryonálního vývoje opět znázorněném na Octomilce obecné) je rozdělen na mnohá odvětví zkoumání (sledování aktivity srdce, sledování cirkadiálních rytmů, sledování paměťového učení, sledování pohybové aktivity), které tvoří celkové střípky mozaiky, která v budoucnu může pomoci k cenným objevům, jako jsou léky na diabetes, Alzheimerovu chorobu a jiné genetické onemocnění, která sužují po dlouhou dobu lidskou populaci.

Samotný výzkum by však nebyl možný či velmi obtížný bez použití speciálních přesných přístrojů, moderní výpočetní techniky a sofistikovaného programového vybavení, které toto zkoumání umožňují a usnadňují. Tento fakt poskytl dostatečný hybný krok pro úzkou spolupráci genetiků, přírodovědců a konstruktérů, programátorů těchto speciálně zaměřených přístrojů. Spolupráce těchto oborů poskytuje výhody pro obě strany se společným cílem – zlepšení kvality života lidí.

Ve své práci vystupuji v roli onoho konstruktéra a programátora, který spolupracuje s pracovníky Biologického centra Akademie věd České republiky na stavbě přístroje na sledování a vyhodnocování pohybové aktivity hmyzu, který by měl biologům usnadnit zkoumání pohybové aktivity (kinematické a dynamické parametry pohybu) Octomilek obecných (*Drosophila Melangostar*). Celý projekt popisovaný v rámci mé diplomové práce je velmi komplexní činností, jejíž příprava, vývoj a samotné provedení přesahuje rámec standardní diplomové práce.



## 2 METODY A SOFTWARE PRO MĚŘENÍ POHYBOVÉ AKTIVITY HMYZU

Na trhu existuje spousta sestav pro měření pohybové aktivity hmyzu. Já jsem se ve své rešerši, potažmo práci zaměřil na tři největší sestavy, z nichž každá přináší do oboru měření pohybové aktivity hmyzu jiné metody a je cílena trochu jiným směrem a pokusil jsem se vyzdvihnout jejich klady a zápory vzhledem k mé plánované konstrukci HW a SW pro měření aktivity hmyzu. Srovnáním těchto stávajících metod a software bylo možné zkonstruovat hardware a software, který je kompromisem mezi všemi popisovanými sestavami. Při rešerši jsem se zaměřil na tyto aspekty řešení:

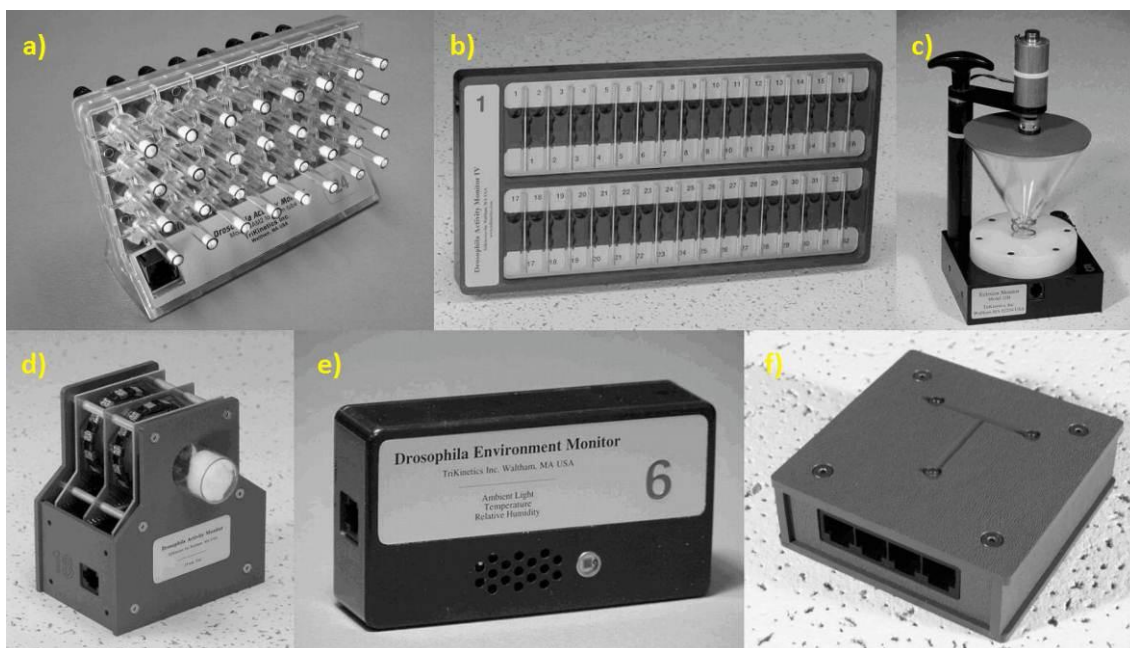
- Schopnosti a víceúčelovost hardwarové části sestavy (pokud je dostupná)
- Schopnosti dodávaného software a jeho nároky na PC
- Pořizovací cena software a případného HW
- Klady a zápory celkového řešení sestavy

### 2.1 TriKinetics

#### 2.1.1 Popis systému

Tento komplexní systém je zaměřen na monitorování pohybové aktivity, charakteru sociálních interakcí a cirkadiálních rytmů u ovocné mušky (*Drosophila*) či podobně rozměrného hmyzu. Systém je schopen biologům poskytnout informace, které přesně charakterizují pohybovou aktivitu hmyzu a to vše na základě jednoduchých pohybových rovnic, které systém v průběhu času sleduje a počítá v reálném čase. Jednotlivé mouchy jsou uzavřeny ve skleněných trubičkách, jež omezí jejich pohyb, který je možné poté ve speciálním stojanu detekovat pomocí infračervených paprsků (IR závor), jež jsou rozmístěny v několika osách po celém obvodu trubičky (zpravidla jde o nastavení, kdy jsou IR závory umístěny v každé 1/3 délky, odtud název TriKinetics). Při průletu mouchy skrze infračervenou závoru přeruší moucha její paprsek (závory jsou umístěny v průřezu v několika osách z důvodu eliminace chyby; čili moucha může prolétat blíže k okraji trubičky), tento stav je pak detekován jako logická hodnota (číslo) konkrétní trubičky (např. v případě tří IR závor jde o hodnotu „010“ – moucha se nachází nebo právě prolétla prostředkem trubičky). Tyto logické hodnoty (čísla) pak jsou periodicky sbírány připojeným měřicím počítačem s dodávaným softwarem DAMS. [1]

Celá sestava se skládá z několika modulů (HW část) a samotného software. Jednotlivé moduly plní vždy jinou funkci a nejsou tudíž univerzální. Jednotlivé části jsou dle manuálu dodávaného k sestavě TriKinetics popsány takto:



Obr. č. 1 Moduly systému TriKinetics: a) Monitor lokomoce se zdvojenými IR závorami; b) Monitor lokomoce s jednoduchými závorami; c) Monitor narození; d) Monitor populace; e) Monitor prostředí; f) Napájecí zdroj a řídicí jednotka systému; převzato a upraveno z [2]

### 2.1.2 Popis jednotlivých částí [1]

- **Inkubátor** (Speciálně upravený pro externí řízení osvětlení pomocí modulu, obsahuje regulátor teploty a vlhkosti s časovou hysterezí)
- **Napájecí zdroj** (Poskytuje napájení až pro 120 modulů přes standardní telefonní kabel, umožňuje zapínání a vypínání osvětlení nezávisle až v 6 inkubátorech naráz pomocí dodávaného software, dále je to hlavní centrum pro sběr dat, zdroj je spojen s PC pomocí USB či RS232), zobrazen na obr. č. 1f)
- **Monitor lokomoce** (32 skleněných trubiček o rozměrech 5,7 nebo 10 mm, které mají přesně ve své půlce umístěnou IR závoru, která umožňuje detekci průletu mouchy z jednoho konce trubičky na druhý), zobrazen na obr. č. 1a),1b)
- **Monitor narození** (skládá se z plastového víka umístěného na nálevce o rozměru 100 mm, do které padají narozené mouchy vylíhlé z kukly, v hrdle nálevky je pak umístěna infračervená závoru, která umožňuje počítat vylíhlé jedince), zobrazen na obr. č. 1c)
- **Monitor populace** (Skleněná trubička pro více jak jednu mouchu o průměru 25 mm rozmanité délky po jejímž celém obvodu jsou umístěny kroužky s infračervenými závorami), zobrazen na obr. č. 1d)
- **Monitor prostředí** (Jde o HW modul s čidly, který měří a kontinuálně zaznamenává teplotu, relativní vlhkost a intenzitu osvětlení v inkubátoru, kde se celé měření provádí), zobrazen na obr. č. 1e)
- **DAMS Software** (Zkratka pochází z anglických slov Data-Acquisition-Monitoring-System). Jde o kolekci cíleně zaměřených software, tvořících však

jeden produkt, pro jednotlivé moduly, který umožňuje archivaci dat z měření po dobu až několika dnů.

### 2.1.3 Schopnosti HW a nároky SW

TriKinetics systém dosahuje vysoké životnosti díky své robustní a částečně mechanické konstrukci. Je vybaven jak starším (RS232) tak i rozšířeným rozhraním USB 2.0 pro sběr dat. Samotný zdroj napájení TriKinetics vyžaduje ještě externí zdroj stejnosměrného napětí 9 V, ze kterého zásobuje elektrickou energií jednotlivé moduly. K systému může být připojeno až 120 monitorů (HW modulů) zároveň, přičemž data je PC schopno sbírat v rozmezí intervalů 1 s až 1 h, vše je omezeno pouze počtem dat, 1 monitor (modul) může poskytnout až 100 000 měření. Systém řízení osvětlení v inkubátoru je čistě programově, respektive uživatelsky řízený a lze jej naplánovat na libovolný čas a datum. [2]

Software DAMS se dělí na několik částí, první je samotný program pro zaznamenávání dat z napájecí části systému, který existuje ve dvou verzích (pro jedno paprskovou IR detekci, pro více paprskovou IR detekci), dále pak tzv. „File Scan software“, který umožňuje filtraci dat z určitých časových úseků a vybraných modulů včetně separace jednotlivých trubiček. Nároky na software DAMS jsou velice nízké, sběr dat je realizován pomocí sériových rozhraní (RS232, USB), nedochází k žádným výpočtům, pouze k ukládání a sběru dat, tudíž nároky na procesor PC a velikost operační paměti jsou extrémně nízké. Samotný software DAMS vyžaduje ke svému běhu operační systém Windows 98 a vyšší (Windows XP) nebo operační systém Macintosh OSX. V době psaní mé diplomové práce byla dostupná na oficiálních stránkách [www.trikinetics.com](http://www.trikinetics.com), poslední dostupná verze DAMSystem 3.03. [2]

### 2.1.4 Pořizovací náklady

Samotný napájecí zdroj (datové centrum) včetně kabeláže a externího 9 V napájecího zdroje lze pořídit dle oficiálního ceníku (dostupného na [www.trikinetics.com](http://www.trikinetics.com)) za \$385 (odpovídá cca 8000 Kč), modul monitoru lokomoce (ve verzi 32 trubiček, dvojitě IR závory, 5 mm délka trubiček) za \$600 (odpovídá cca 12000 Kč) dále pak monitor prostředí za \$450 (odpovídá cca 9000 Kč), ostatní moduly se pohybují cenově nad hladinou \$700 (cena převyšuje cca 13000 Kč), čili provozně schopný systém pro měření lokomoce (v rámci mé diplomové práce objekt zájmu) je možné pořídit za celkovou cenu  $\$385+\$600+\$450=\$1435$  (odpovídá cca 28000 Kč). Software DAMS je možné stáhnout z internetu zdarma a taktéž jej provozovat. [3].

(\* Přepočten cen na české koruny byl kalkulován na základě kurzu dolaru k datu 24. 12. 2011, který činil 19,80 Kč / dolar, zdroj [4])

### 2.1.5 Klady a zápory

Z kladů bych vyzdvihl dlouhodobý záznam jednoduchých dat (až po dobu několika dní), dlouho ověřený byť již dnes zastaralý systém měření pohybové aktivity. Možnost měření vysokého počtu paralelních jedinců. Snadné a jednoduché zaškolení personálu obsluhy sestavy, data jsou zpracovávána v reálném čase, avšak bez možnosti zpětné kontroly či reprodukce.

V případě záporů bych upozornil na to, že pozice mouchy není sledována v celém rozsahu jejího možného pohybu (jsou detekovány pouze průchody skrze infračervené závory) = vznikají tak místa kde moucha není vůbec sledována (v prostoru mezi infračervenými závorami), poměrně vysoké pořizovací náklady, nutnost manuální a velmi složité sterilizace jednotlivých trubiček [5]. Nemožnost rozlišení jednotlivých identit much (není známo, která moucha v trubičce přerušila infračervený paprsek závory). Software již není dlouho upgradován (naposledy v roce 2008)

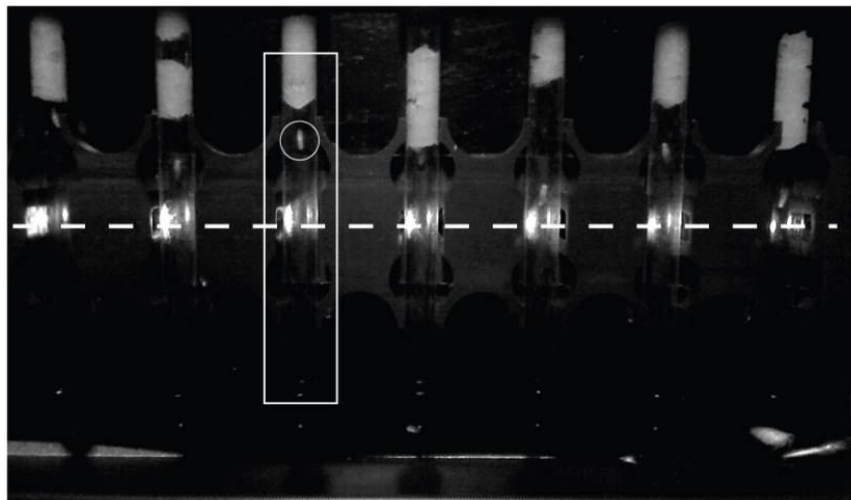
## 2.2 PySolo

### 2.2.1 Popis systému

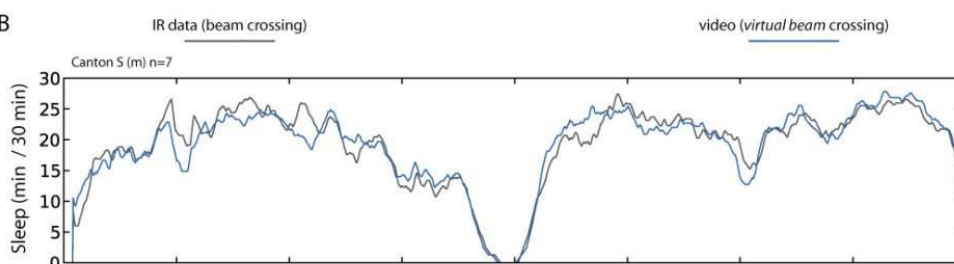
Jde o komplexní software autorů Giorgia F. Gillestra a Chiara Cirelliho z Univerzity ve Wisconsinu v Madisonu (USA), který slouží pro analýzu spánku a aktivity ovocných mušek *Drosophila*. Je zaměřen speciálně na bezobratlé živočichy drobné velikosti, tak, že výsledky poskytuje v přehledné formě a to vše zcela zdarma. Umožňuje jak sběr dat ve formě natáčení videa a snímání obrázků, tak i samotnou analýzu a zpracování dat, umožňuje import dat z předchozího systému TriKinetics tak i samostatné pořízení dat, v tomto případě je nutné funkční obrazově záznamové zařízení (např. webovou kameru). Snahou tohoto SW je kontinuální sledování pohybové aktivity much, čili odstranění hlavního nedostatku systému TriKinetics (ten mouchy dokáže sledovat pouze po přechodu IR závorou), tento neduh se snaží realizovat hledáním pozice mouchy ve videu s intervalem 5s a porovnat nalezenou vzdálenost oproti virtuální IR závoře a tím získat výslednou aktivitu a eliminovat nezměřené a tudíž i neanalyzované místo (mezi IR závorami) jako je tomu v případě DAMS. Tento systém je také schopen emulovat softwarovou cestou chování systému TriKinetics (virtuální IR závory), je tak možné dokázat odchylky při měření pomocí DAMS a Pysolo, jak provedli autoři programu Pysolo, což je znázorněno na obr. č. 2 (jsou zde odchylky při měření aktivity kontinuálním systémem Pysola a virtuální emulací TriKinetics systému). [6]

Vstupní data pro tento software tvoří buď video nahrávané v reálném čase pomocí externího programu (jsou nutné knihovny v operačním systému) či výstupní data v defaultním formátu poskytovaném systémem DAMS, tyto data je však nutné importovat do programu. Tyto data však již neobsahují potřebné informace a materiály pro následnou rekonstrukci pořízeného záznamu který byl podroben analýze.

A



B



Obr. č. 2 a) Náhled detekovaného snímku videa, moucha je detekována i mimo virtuální IR paprsek (naznačení přerušovanou čarou); b) Výstupní graf software Pysolo ukazující porovnání detekčních metod systému DAMS a Pysolo (modrá – kontinuální systém Pysolo s virtuálními IR závorami, šedivá – systém DAMS s fyzickými IR závorami); převzato a upraveno z [7]

## 2.2.2 Schopnosti HW a nároky SW

Software je multiplatformní, dostupný pro operační systémy Linux i Windows a je dodáván pod licencí GNU GPL, jde tedy o svobodný software. Jeho nároky na systém jsou středně velké (uváděno jako středně silné PC v případě analýzy videa v reálném čase – omezeno 5 FPS), SW vyžaduje pro svůj běh operační systém Windows 2000/XP/Vista/7 nebo Linux (verze jádra 2.6.35 či vyšší a verze Python 2.6). Díky video rozšíření, distribuovanému na oficiálních stránkách [www.pysolo.net](http://www.pysolo.net), je možné pořizovat pomocí hlavního SW obrazový záznam pro analýzu v reálném čase, čili bych uvedl jako pomocný HW webovou kameru s patřičnou úpravou filtru objektivu (návod je zveřejněn taktéž na oficiální internetové stránce, úpravou webové kamery však dochází ke ztrátě záruky), poněvadž video i samotný program počítají s kompozicí světlé mouchy a tmavého pozadí (mouchy jsou osvětleny IR LED a světélkují, díky IR filtru na kameře je obraz víceméně černobílý a mouchy se tudíž jeví jako bílá a pozadí černé). Samotné výsledky programu pak obhospodařuje volně šiřitelná knihovna Pythonu (existuje i ve své multiplatformní verzi) pro tvorbu 2D i 3D grafů „matplotlib“, která také slouží více známému matematickému programu MATLAB jako engine pro tvorbu grafů (knihovnu vytvořilo studio Wolfram Research, Inc.) [8]

### 2.2.3 Pořizovací náklady

Díky svobodné licenci a možnosti použít volně a zdarma šiřitelný operační systém Linux jsou pořizovací náklady na software nulové. Avšak je nutné získat záznamové zařízení (pokud nejsou používána data ze systému TriKinetics), čili náklady vzrostou na pořizovací cenu webové kamery.

### 2.2.4 Klady a zápory

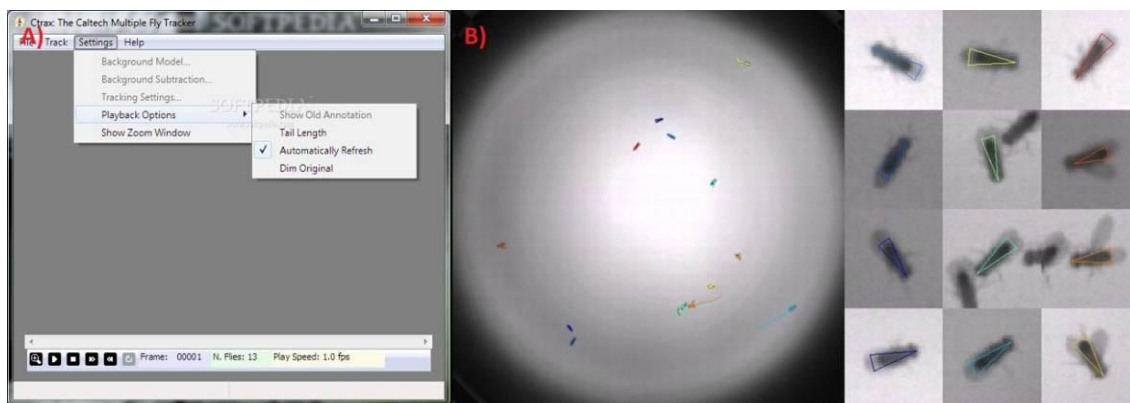
Mezi klady hodnotím implementaci knihoven NumPy (<http://numpy.scipy.org/>) a SciPy(<http://www.scipy.org/>) do programu Pysolo, tyto knihovny (součásti jazyka Python) umožňují velmi rychlé výpočty v reálném čase (jsou z velké části psány napsány ve strojovém jazyce PC) a umožňují velmi rychlé zpracování dat, avšak jejich absence v defaultně nainstalovaném systému, vyžaduje jejich dodatečnou distribuci s programem, z čehož plyne větší velikost instalačního balíčku. Program dále disponuje takřka dokonalou reprezentací výsledných dat.

Mezi zápory bych rozhodně zařadil poměrně problematický chod programu v OS Linux (jsou potřebné vždy určité verze referenčních knihoven a samotného Pythonu). Dále pak poměrně nepřehledné a složité grafické rozhraní programu, které je složeno z databázové části, části analyzační a části která je schopna pořídít obrazový záznam, to pak způsobuje poměrně hluboké zaškolení obsluhujícího personálu.

## 2.3 CTRAX

### 2.3.1 Popis systému

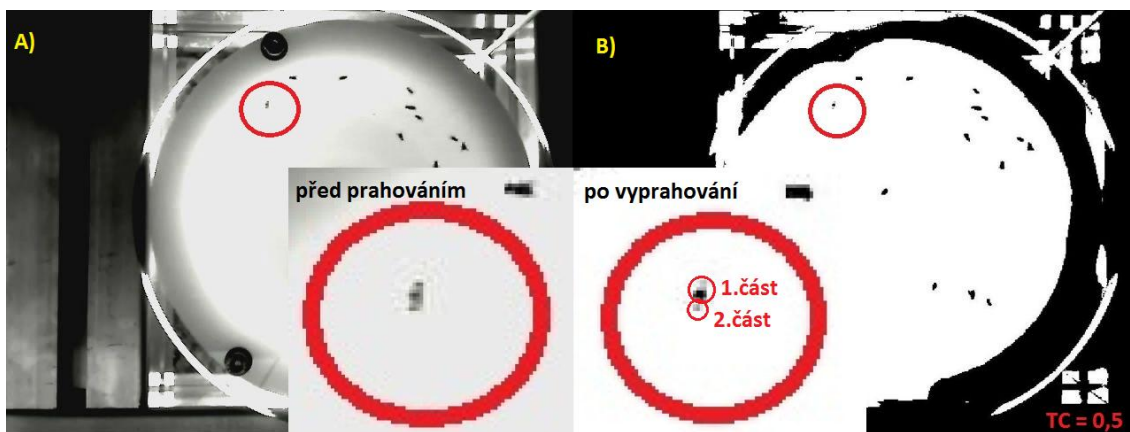
Ctrax je volně šiřitelný a dostupný (open-source) sofistikovaný software ([www. http://ctrax.sourceforge.net/index.html](http://ctrax.sourceforge.net/index.html)), od americké programátorky a autorky Kristin Branson, Ph.D., který je uzpůsoben k trasování roje ovocných mušek a k udržování jejich individuálních identit po co nejdelší dobu letu (dle oficiálních stránek až po dobu 1,5 h). Pro eliminaci chyb je software vybaven externí knihovnou „FixErrors Matlab GUI s grafickým rozhraním pro MATLAB, která zajistí detekci špatně nalezených identit much a umožní je tak snadno a jednoduše opravit. Dále disponuje mnoha skripty „Behavioral Microarray Matlab Toolbox“, které umožňují rozličné možnosti výstupních dat (grafů a parametrů) dle zaměření na chování a aktivitu much. Výsledkem software je poskytnout biologům kvantitativní výsledky chování (pozice, rychlost, zrychlení, otočení) a neurologické výsledky mozkových pochodů (místa kde se mouchy nejčastěji shlukují, pozice ve kterých se shlukují apod.) volně se pohybujících much (v rámci omezujícího prostoru misky či trubičky), jak je vidět na obr. č. 3, za účelem archivace těchto dat a následnou tvorbou postulátů a rovnic pro popis chování a zkoumání funkce mozku těchto much. [9]



Obr. č. 3 a) Náhled programu CTRAX, převzato a upraveno z [10]; b) Náhled výstupního videa s detekovanými mouchami, převzato a upraveno z [11]

### 2.3.2 Schopnosti HW a nároky SW

Detekční schopnosti a úspěšnost detekce programu jsou velmi dobré, smysl detekce je založen na výpočtu ideálního pozadí (čili tvorbě virtuálního snímku bez much) který je pak použit při odečítání od snímku z reálného videa (nacházejí se na něm mouchy). Výsledek odečtu (rozdílový snímek) pak obsahuje samotné mouchy, které je možné detekovat. Tento systém je vysoce funkční za předpokladu, že video je pořízeno s minimálním šumem (obvykle je však video natáčeno za špatných světelných podmínek), jelikož pak šum ovlivňuje barevné složky pixelů, které tvoří obraz mouchy, může dojít i po odečtu a vyprahování obrazu k rozpadu mouchy na více částí na což detektor softwaru zareaguje nalezením dvou much místo jedné (viz. obr. č. 4). [12]



Obr. č. 4 Zobrazení simulace detekčního bugu vstupního videa s příliš velkým šumem v obraze (nastavení koeficientů prahování v rozmezí 0 až 0,5); A) Náhled snímku z videa s detailem problematické mouchy; B) Náhled vyprahovaného snímku z videa s detailem rozpadu problematické mouchy na dvě části.

Software CTRAX není schopen sám o sobě spolupracovat s obrazově záznamovým zařízením (kamerou atd.), jde tedy o software, jehož vstupem jsou již pořízená data (video), čili analýza neprobíhá v reálném čase. Samotný program je založen na programovací platformě Python a na mnoha závislých knihovnách. Vše je

však dostupné jako open-source, a je takřka nezávislé na operačním systému (verze pro Linux, Unix, Mac OSX ale i pro Windows), jak v 32bitové tak i v 64bitové verzi.

Jak uvádí oficiální zdroj [12], HW nároky programu nejsou stanoveny, na pomalých PC běží program pomaleji než na rychlých PC s výkonnými procesory, program však není multiprocesorový. Nároky na zvýšené použití operační paměti PC jsou vyžadovány pouze u velkých nekomprimovaných videí a při exportu a ukládání výsledných dat ve formátu \*.mat (pro MATLAB).

Vstupními daty programu jsou tedy pořízená videa ve formátech (MPEG-1, MPEG-2, MPEG-4, HUFFYUV, nekomprimované AVI, H-264 a mnoho dalších), doporučený formát vstupních dat je však buď nekomprimované AVI (s nastavením luminance Y8, 24 bit RGB barevnou hloubkou a konstantním datovým tokem videa, paradoxně však program CTRAX ignoruje barevné vzezření videa a veškeré snímky převádí do odstínů šedi) nebo formát podporovaný FFmpeg knihovnamí (<http://www.ffmpeg.org/general.html#SEC3>) s konstantním datovým tokem (proměnný datový tok není schopen program CTRAX korektně detekovat). [12]

Standartním výstupním formátem programu CTRAX je soubor s příponou \*.ann, jež obsahuje krom samotné hlavičky (zdroj videa, FPS videa, data pro rekonstrukci virtuálního pozadí bez objektů) i samotné pozice (X, Y) detekovaných much včetně parametrů elips, které je obklopují, dále úhly otočení jednotlivých much, čísla jednotlivých identit much a velikosti jednotlivých much v pixelech. [13]

### 2.3.3 Pořizovací náklady

Pořizovací náklady na SW jsou díky jeho šíření pod licenci open-source nulové. Avšak díky úzké provázanosti na program MATLAB při následném zpracování dat je nutné vlastnit jeho licenci, což není levnou záležitostí (pro komerční účely se cena pohybuje kolem 60000 Kč, v případě školní licence pak kolem 18 000 Kč; dle oficiálního českého distributora [14] k datu 2. 9. 2011). V poslední verzi programu CTRAX ve verzi 0.3.0 je možné vygenerovat jednoduchý náhled kalkulovaných výsledků, avšak ten nepostačuje ani zdaleka schopnostem externích knihoven (skriptů) MATLAB dodávaných k programu.

### 2.3.4 Klady a zápory

Hlavní devizou a kladem programu je jeho relativní rychlost zpracování dat, avšak vzhledem k tomu, že se program až příliš spoléhá na následnou uživatelskou eliminaci chyb pomocí nástrojů dodávaných ve formě knihoven s grafickým rozhraním pro MATLAB, je detektor programu velmi omezený a poskytuje leckdy mnohdy nesprávná data.

Mezi největší zápory bych zařadil spoustu chyb v programu (bugů), software není schopen korektně z hlavičky AVI soubory zjistit informace o FPS (snímkové frekvenci videa), tudíž není možné korektně kalkulovat rychlost či zrychlení objektu



(mouchy). Dále jsem při svém testování narazil na poměrně nepohodlné řešení knihoven pro eliminace nekorektně identifikovaných much, samotné knihovny fungují velmi dobře, avšak je nutné přednastavit parametry detekce tak, aby ony chybné detekce byly vůbec nalezeny, nehledě na to, že je potřeba zapotřebí zakoupená licence komerčního software MATLAB pro další zpracování dat z Ctrax (eliminace chyb, analýza chování much atd.). Program neumožňuje žádný export dat do formátů použitelných softwaru třetích stran, čili jsou veškerá data zapouzdřena do té doby, než je uživatel otevře v programu MATLAB (odtud už jdou exportovat jako XLS formát pro MS Excel).

Program má dále velké problémy s překrývajícími se mouchami (dotýkající se mouchy je schopen variabilním krokováním prahovacího koeficientu separovat), což řeší oficiální prezentace videí konstrukcí misek takových, kde k tomuto nedochází (snížena výška misek na úroveň výšky mouchy)

## 2.4 Parametry plánované konstrukce přístroje

Zobecněním všech parametrů výše uvedených software a plánovaným účelem mého software a hardware jsem dospěl k následujícím závěrům, které by měli stručně a obecně popisovat funkci a schopnosti mého zařízení:

### Parametry HW:

- Zařízení musí obsahovat vlastní kvalitní obrazově záznamové zařízení (webovou kameru) s dostatečně dimenzovanou rezervou a možností úprav tak, aby bylo možné pořizovat záznam přizpůsobený konkrétním účelům
- konstrukce musí být dostatečně variabilní a flexibilní pro rozličné prostory, kde budou mouchy uzavřeny (trubičky, misky)
- Pořizovací cena konstrukce bude co nejnížší, ale přitom dostatečně tuhá a mobilní.

### Parametry SW:

- Program bude zcela nezávislý na knihovnách třetích stran, nebude vyžadovat SW třetích stran (zcela vylučuji komerční sféru)
- Výstupní data programu budou dostupná v takových formátech, jejichž mateřské programy jsou nejrozšířenější pro konkrétní operační platformu systému (Windows)
- Program bude šiřitelný zdarma (mimo zdrojových kódů)
- Program bude jednoduchý na obsluhu s přehledným grafickým rozhraním

## Použité zdroje

- [1] Trikinetics: Catalog 1105 [online]. c2005 [cit. 2011-10-27]. Dostupné z World Wide Web: <<http://www.trikinetics.com/Downloads/Catalog1105.pdf>>
- [2] Trikinetics: DAMSystem User's Guide 3.0 [online]. c2007 [cit. 2011-10-27]. Dostupné z World Wide Web: <<http://www.trikinetics.com/Downloads/DAMSystem%20User's%20Guide%203.0.pdf>>
- [3] Trikinetics: Price list [online]. c2010 [cit. 2011-12-27]. Dostupné z World Wide Web: <<http://www.trikinetics.com/Downloads/TriKinetics%20Price%20List.pdf>>
- [4] Kurzy.cz: Americký dolar USD [online]. c2011-12-24 [cit. 2011-10-24]. Dostupné na World Wide Web: <<http://www.kurzy.cz/kurzy-men/nejlepsi-kurzy/USD-americky-dolar/>>
- [5] JOVE: Assaying locomotor activity to study circadian rhythms and sleep parameters in drosophila [online]. c2009-09-28 [cit. 2011-10-27]. Dostupné z World Wide Web: <<http://www.jove.com/video/2157/assaying-locomotor-activity-to-study-circadian-rhythms-and-sleep-parameters-in-drosophila>>
- [6] PySolo: Features[online]. c2009-04-15 [cit. 2011-10-27]. Dostupné na World Wide Web: <<http://www.pysolo.net/features/>>
- [7] PySolo: Video validation [online]. [cit. 2011-12-27]. Dostupné na World Wide Web: <[http://www.pysolo.net/files/support/video\\_validation/Figure1.jpg](http://www.pysolo.net/files/support/video_validation/Figure1.jpg)>
- [8] Pysolo: General Requirements [online]. c2011-07-28 [cit. 2011-10-24]. Dostupné na World Wide Web: <<http://www.pysolo.net/docs/general-requirements/>>
- [9] CTRAX: The Caltech Multiple Walking Fly Tracker [online]. c2011 [cit. 2011-10-27]. Dostupné na World Wide Web: <<http://ctrax.sourceforge.net/>>
- [10] Ctrax: The Caltech Multiple Fly Tracker [online]. [cit. 2011-10-27]. Dostupné na World Wide Web: <[http://i1-win.softpedia-static.com/screenshots/Ctrax\\_2.png](http://i1-win.softpedia-static.com/screenshots/Ctrax_2.png)>
- [11] CTRAX: Detection of flies [online]. [cit. 2011-10-27]. Dostupné na World Wide Web: <[http://ctrax.sourceforge.net/images/ctrax\\_012.png](http://ctrax.sourceforge.net/images/ctrax_012.png)>
- [12] CTRAX: Computer and software requirements [online]. [cit. 2011-10-27]. Dostupné na World Wide Web: <<http://ctrax.sourceforge.net/install.html#computer-and-software-requirements>>
- [13] CTRAX: Annotation file format[online]. [cit. 2011-10-27]. Dostupné na World Wide Web: <<http://ctrax.sourceforge.net/ctrax.html#annotation-file-format>>
- [14] Humusoft: Ceník produktů systému MATLAB [online]. [cit. 2011-10-27]. Dostupné na World Wide Web: <<http://www.humusoft.cz/produkty/matlab/cenik/matlab.pdf>>

## 3 HW ŘEŠENÍ PŘÍSTROJE

### 3.1 Požadavky na konstrukci

Již zpočátku stavby modelu přístroje na sledování a vyhodnocování pohybové aktivity hmyzu bylo zřejmé, že díky dostupné ceně profesionálně vyrobeného zařízení, nebude nutné elektronicky konstruovat samotnou část zařízení odpovědnou za přenos obrazového materiálu do PC (kamera), mohl jsem se tedy zcela zaměřit na nosnou část konstrukce, která bude plnit funkci stojanu a příslušenství, které bude stojan (konstrukci samotného přístroje) doplňovat (osvětlení apod.).

Idea návrhu byla následující, nosná konstrukce složená z masivních profilovaných tyčí bude tvořit kostru přístroje. Nosné tyče musejí být z takového materiálu, který bude dostatečně tuhý (nebude se kroutit), bude lehký (pro snadnou manipulaci a přemísťování celé konstrukce) a samotné tyče, ze kterých bude konstrukce složena, budou mít ve svém průřezu profil umožňující snadnou montáž konstrukce (žlábký pro montážní šroubky). Kostra bude mít několik pater, které budou plnit různé funkce. První (nejspodnější) patro bude plnit funkci uchycení osvětlení pro druhé (prostřední) patro, kde budou uchyceny skleněné omezující prostory pro mouchy (trubičky, misky apod.). Děj uvnitř misek či trubiček bude snímat kamera umístěna ve třetím patře. Všechna patra budou v rámci konstrukce mobilní a bude s nimi možno manipulovat (snadno posunovat nahoru a dolů dle potřeby). Bude možné přidávat a ubírat libovolné množství pater dle potřeby.

### 3.2 Návrh a model konstrukce

#### 3.2.1 Konstrukční tyče

Konstrukční profilované tyče, použité na mou konstrukci, pocházejí z nosných konstrukcí pro upevnění fotovoltaických panelů, jsou zobrazeny na obr. č. 1. Výhodou těchto tyčí je jejich materiál, ze kterého jsou vyrobeny (lehká a zároveň odolná slitina hliníku) a zároveň jsou připraveny pro možnost snadné montáže (jsou profilované pro různé typy uchycení, já tento profil využil pro uchycení šroubů s maticemi). Díky velkovýrobě bylo možné tyče pořídit v různých délkách, což mi umožnilo volnou ruku při návrhu konstrukce. Samotné tyče jsem byl nucen opatřit matným černým nátěrem z důvodu nežádoucích odlesků, jelikož materiál, ze kterého byli vyrobeny, byl extrémně lesklý a odrážel by světlo z podsvícení konstrukce.



Obr. č. 1 a) Profil konstrukčních tyčí; b) Pohled na profilované montážní úchyty konstrukčních tyčí; c) Finální sestava konstrukčních tyčí s naneseným matným černým nátěrem včetně klínů a spojek

### 3.2.2 Podsvícení konstrukce

Osvětlení musí splňovat následující požadavky. Musí být dostatečně velké (v šíři konstrukce, aby pokrylo světelným kuzelem oblast, kde se budou nacházet misky či trubičky s mouchami). Stínítko osvětlení musí mít takové vlastnosti (mléčné sklo), aby co nejvíce homogenizovalo světlo ze zdroje, zamezí se tak přesvícení částí obrazu při snímání pomocí kamery.

Prvotní záměr použít klasické žárovky jsem zamítl, bylo by nutné komplikovaně odvádět přebytečné teplo, které generují. Paradoxně by však pro mouchy šlo o zřejmě nejpřirozenější světelný zdroj, spektrum elektromagnetického záření, které žárovka

vyzařuje je totiž velmi podobné slunečnímu záření. Proto jsem přistoupil ke svítidlům na bázi LED diod, které negenerují nežádoucí teplo. Dle mých testů dokázala dostatečně misky (trubičky) světlem zásobovat již 5 W LED žárovka, vybral jsem proto předdimenzované 10 W LED svítidlo se 145 LED diodami o průměru 14 cm se závitem E27 (objednáno z <http://www.dealextreme.com>) pro osvětlení všech misek (plánovány 4) a trubiček (plánováno 8 kusů). Avšak díky kruhovému tvaru světla a poměrně úzkému vyzařovacímu úhlu LED diod došlo k tomu, že při použití všech dostupných misek či trubiček, byl obraz poměrně nehomogenně nasvícen a to i při pokusy se dvěma stejnými zdroji světla a deskami z mléčného skla.

Dlouhým testováním a zkoušením jsem proto finálně pro podsvícení konstrukce vybral velké interiérové kruhové svítidlo o průměru 30 cm obsahující 3 kompaktní úsporné zářivky s patičí E27 (příkon 20 W, svítivostí jde o ekvivalent 100 W klasické žárovky) pootočené osově o 120°, jehož stínítka bylo vyrobeno z tlustého mléčného skla jak je vidět na obr. č. 2.



**Obr. č. 2 a) Interiérové svítidlo použité pro konstrukci ve vypnutém stavu; b) Interiérové svítidlo ve svém aktivním stavu bez použití dodatečného mléčného skla (patrná malá homogenizace světla; c) Interiérové svítidlo s použitím dodatečného mléčného skla, které světlo dokonale rozptýlí**

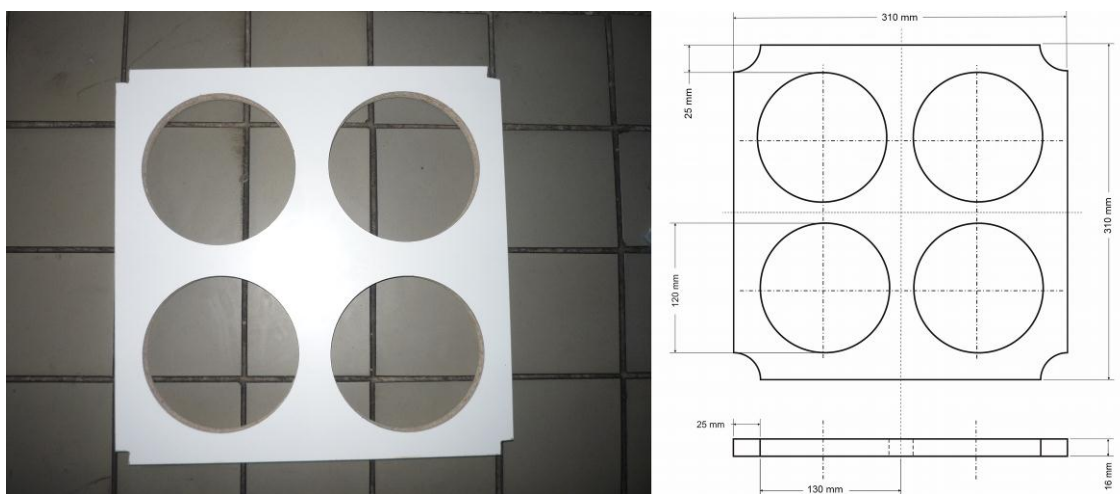
Samotné zářivky blikají díky síťové frekvenci 50 Hz na dvojnásobné frekvenci (světelného maxima totiž dosahují jak v kladné tak záporné hodnotě síťového napětí), čili na 100 Hz. Tato frekvence je dle konzultací s biology tak vysoká, že ji nejsou schopny mouchy zaregistrovat a neměla by ovlivňovat měření (v budoucnu je však nutné tuto premisu dále ověřit).

Samotné mléčné sklo nedokázalo spolehlivě a dostatečně homogenizovat světlo, bylo tedy nutné světlo dodatečně homogenizovat, to jsem provedl pomocí 3 mm mléčného plexiskla umístěného 15 cm nad úroveň stínítka světelného zdroje, vliv homogenizace pomocí tohoto skla je zobrazen na obr. č. 2c).

### 3.2.3 Plato pro misky a trubičky

Na konstrukci bylo potřeba přidělat i držáky (formy) pro skleněné misky a trubičky, ve kterých se budou nacházet mouchy. Tento problém zpočátku vyřešila sololitová deska, v níž byly vyříznuty 4 otvory pro misky o průměru 12cm (přesah 5 mm skleněných vík misek bránil miskám v propadu skrze sololitovou desku), avšak díky poměrně velké váze všech misek a nízké tuhosti sololitu (tloušťka desky cca 3 mm) se sololit prohýbal, navíc ho nebylo možné užitečným způsobem použít pro uchycení trubiček.

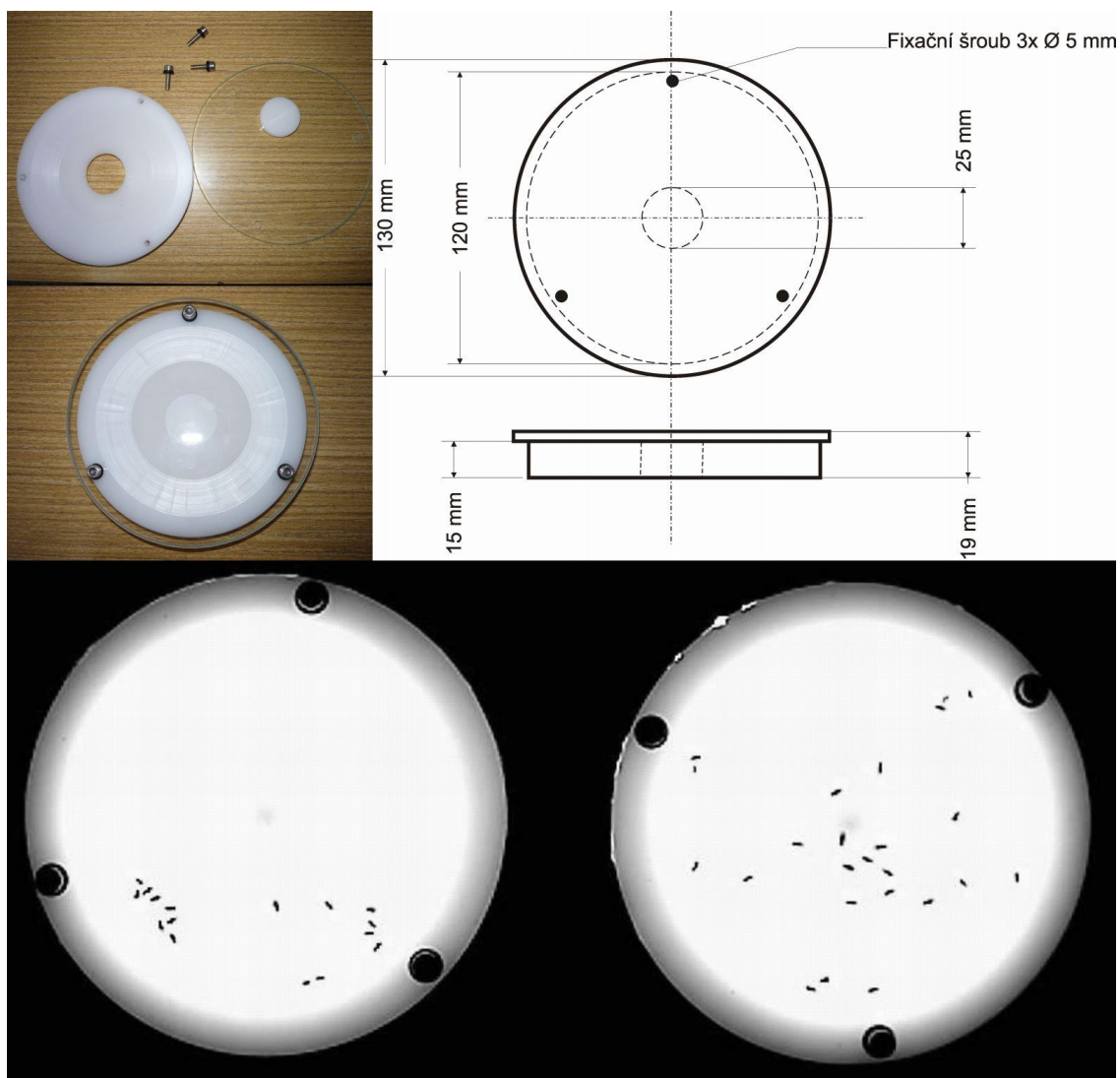
Rozhodl jsem se proto pro výrobu dvou plat, z masivní dřevotřísky (tloušťka 1,6 cm), které poskytnou dostatečný prostor a snadnou manipulaci pro výměnu misek či trubiček.



Obr. č. 3 Vlevo finální plato uzpůsobené pro 4 snímané misky, vpravo technický náčrtek pro výrobu plata s rozměry

Plato pro misky zobrazené na obr. č. 3 je uzpůsobeno pro 4 misky. Díky tloušťce dřevotřískové desky (16 mm), ze které je plato vytvořeno, je možné skrýt celou tloušťku plastové misky (obdobné tloušťky jako plato), tím se eliminuje nežádoucí obrysový světelný kruh (světlo prosvítá mezerami mezi hranou plata a miskou). Manipulace s platem je jednoduchá a lze jej kdykoliv a snadno vyměnit za jiné plato specializované pro jiný účel (trubičky pro měření pohybové aktivity na základě externě a uměle vyvolaného podnětu). Před samotným měřením bylo nutné vždy plastový materiál misky umýt, materiál misky, zejména prostřední demontovatelná a často používaná část (vstupní a výstupní otvor pro mouchy) je totiž velmi náchylný na znečištění a miska by se pak pod homogenním podsvícením jevila jako nehomogenní plocha (prostřední část by byla tmavší). Je samozřejmé, že misku je nutné před použitím umýt speciálními roztoky na bázi alkoholu, eliminují se tak případné nežádoucí látky po předchozích mouchách (feromony, zbytky těl, výměšky atd.), které by mohly silně ovlivnit měření. Na základě prvních testů s mouchami byla výška misky přesněji výška misky v prostoru pro mouchy až příliš velká (bylo potřeba zajistit, aby samice octomilky mohla vztyčit křídla, je totiž větší než sameček), díky tomu se vešly dvě mouchy (samečkové) nad sebe a zneprůjemňovaly tak analýzu. Tento neduh jsem po konzultaci s biologem

eliminovat na minimum pomocí natření skleněného víka misky emulzí silikonu a alkoholu. Miska zobrazená na obr. č. 4 se skládá z plastového těla, skleněného víka s přesahem a třemi fixačními šrouby.

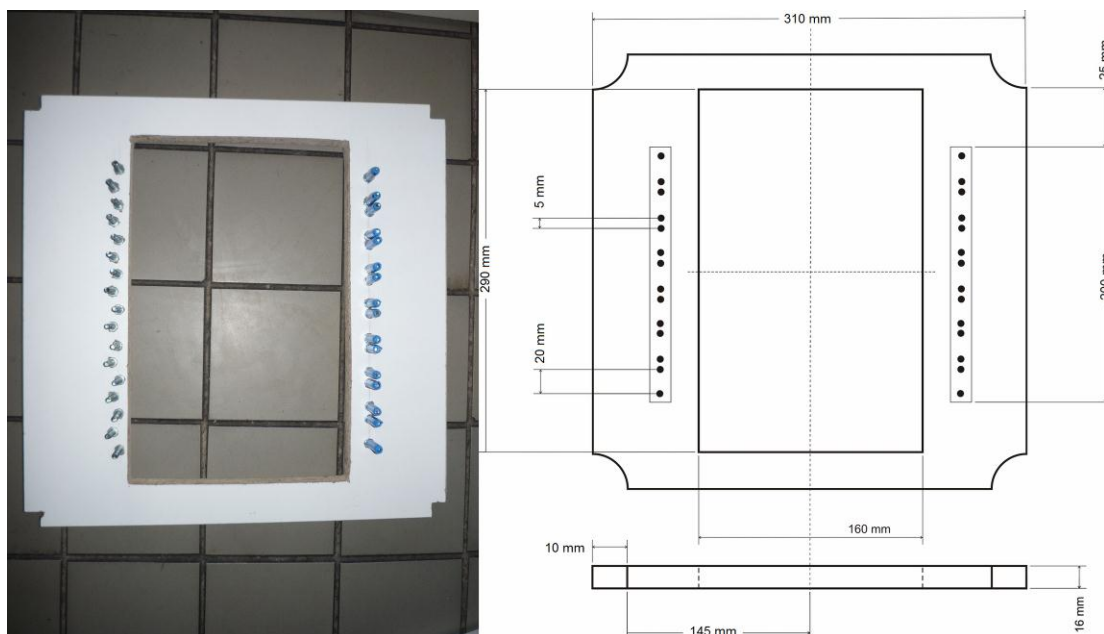


Obr. č. 4 Vlevo nahoře náhled finální misky pro octomilky včetně jejich částí, vpravo nahoře pak technický náčrtek pro výrobu; dole náhled 2 misek při analýze videa.

Druhé plato je zobrazeno na obr. č. 5 a slouží pro zafixování úzkých skleněných trubiček pomocí šroubů (těla šroubů slouží jako vymežovací prostory pro trubičky). Samotné šrouby jsou pak obaleny PVC měkčením z důvodu zabránění pohybu trubiček („viklání v prostoru mezi vymežovacími šrouby“) díky zpětnému rázu, do trubiček je totiž nárazově vháněn vzduch pod tlakem 3 atm. Měkčení je vyrobeno nastříhaných 1,5 cm PVC hadic pro vzduchotechniku.

Díky tomu, že se trubičky nacházely poměrně blízko u sebe (cca 1 cm), bylo nutné zamezit vizuálnímu kontaktu much z jedné trubičky do druhé, aby nebylo měření ovlivněno náhodnými vlivy sousedních much. Vyrobil jsem proto stěny, které oddělují jednotlivé trubičky od sebe. Díky dolnímu podsvícení však bylo nutné stěny upravit tak, aby byly pokud možno co nejúžší a jejich materiál byl lesklý (bylo potřeba rozptýlit

světlo přicházející zespodu). Díky úzkému profilu a lesklému povrchu tak stěny způsobují minimální boční stín u trubiček, s nimiž sousedí (skleněná stěna trubičky se chová jako velmi nevyzpytatelný světlovod). Na výrobu jsem proto použil tvrdý kartonový tenký papír, který jsem obalil silnou hliníkovou fólií, vše je zalepeno a zafixováno proti poškození pomocí průhledné izolepy.



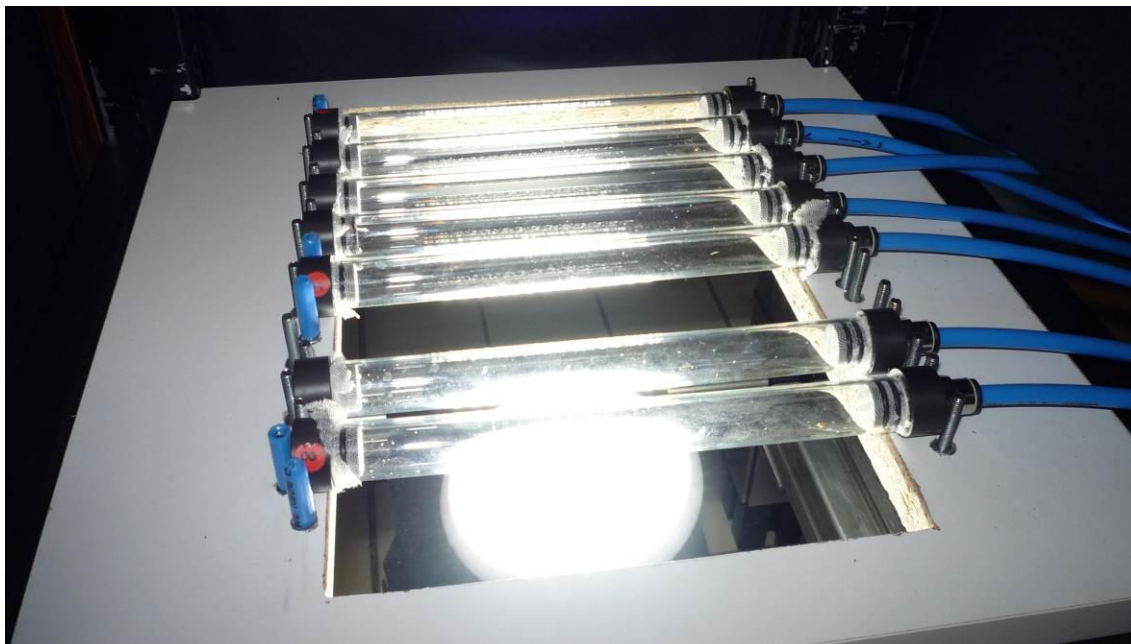
Obr. č. 5 Vlevo náhled plata pro uchycení 8 skleněných trubiček, vpravo pak technický náčrtek pro výrobu.

I zde v tomto případě je nutné před měřením přistoupit k důkladnému čištění trubiček. Samotné trubičky jsou však velmi křehké a manipulaci s nimi vyžaduje zvláštní pozornost (špunty u okrajů trubiček se chovají vůči tenké stěně trubičky jako velmi silná páka a hrozí vylovení okrajů trubičky při nesprávné manipulaci, jsou vyobrazeny na obr. č. 6).

Otvor v místě, kde se nachází trubičky, slouží pro přístup světla, jeho rozměry jsou voleny záměrně, bylo totiž potřeba získat mírnou plochu na platě pro přesahy mléčných skel, které budou už tak homogenizované světlo zespodu konstrukce ještě více rozptylovat. Testováním jsem zjistil, že samotná mléčná skla pod trubičkami nestačí a to ani v různých kombinacích či tloušťkách, přistoupil jsem proto k proložení 2 vrstev skel s mléčným nádechem papírem s velmi jemnou strukturou.

Na obr. č. 6 je znázorněno plato s instalovanými trubičkami. Modré hadice pro vzduchotechniku slouží pro přívod vzduchu přes těsnící špunty na okrajích trubiček. Díky vnitřnímu profilu trubiček nebylo dostatečně možné zajistit optimální průtok vzduchu skrze trubici, proto jsem se rozhodl dovnitř špuntu umístit malý smotek vaty, který náraz proudícího vzduchu rozptýlí a mírně ztlumí (je potřeba korigovat tlak vzduchu přímo na kompresoru).





Obr. č. 6 Plato pro uchycení skleněných trubiček s nainstalovanými trubičkami připojenými k vzduchovému dávkovači

### 3.2.4 Kryt konstrukce

Díky potřebnému podsvícení misek či trubiček, ve kterých se budou nacházet mouchy, bylo nutné zamezit přístupu nežádoucího světla z okolí. Eliminují se tak parazitní odlesky na skleněných částech misek či trubiček, pod kterými by mohli při snímání opticky „mizet“ mouchy. Původní záměr obehnat celou konstrukci sololitovým pouzdem se při stavbě nestala šťastnou volbou, manipulace s částmi konstrukce a její přemísťování se stalo velmi obtížné, proto jsem se rozhodl, že stínítka (na všech čtyřech bocích nosné konstrukce) před okolním světlem musí být z černé neprůhledné látky. Tím pádem stínítka mohlo být kdykoliv „sbaleno“ a „rozbaleno“ dle potřeby, váha látky neovlivnila snadnou manipulaci konstrukce a samotný estetický dojem z konstrukce si velmi polepšil, nehledě na poměrně snadnou výrobu a pořizovací cenu látky. Samotná látka drží na konstrukci na tenké gumičce, jež je protažena horním lemem látky. Lem se nachází i na dolní straně, jelikož byla látka nepoddajná a snadno se mačkala, bylo nutné ji zatížit a napnout ji. Do dolního lemu jsou umístěny dřevěné tyčky jako zátěž.

### 3.2.4 První zhotovení modelu nosné konstrukce

Samotná zhotovená a sestavená konstrukce má poměrně velkou váhu (cca 30 kg) a tuhost což zajistí dostatečnou robustnost proti otřesům. Průřez konstrukce tvoří čtverec o délkách hran 41 cm, výška konstrukce je 75 cm (s ramenem pro kameru pak v závislosti na nastavení výšky kamery cca 100 cm). Konstrukční tyče jsou smontovány pomocí několika plochých klínů s otvory pro šrouby. Spodní část styčných ploch konstrukce a podlahy je opatřena gumovými podložkami pro odstranění nežádoucích

otřesů (chůze obsluhujícího pracovníka apod.). Matným černým nátěrem, kterým byla konstrukce opatřena, je zajištěno, že světlo nezpůsobuje nežádoucí odlesky na lesklých částech misek či trubiček.

Montáž konstrukce je pro jednoho člověka poměrně problematická, je potřeba do drážek jednotlivých konstrukčních tyčí vsadit potřebný počet matek a ty pak samotné vhodným způsobem přemístit na styčnou plochu kde bude dosedat klín se šroubem (M8). Pro přemísťování matek jsem používal slabý imbusový klíč, který do drážek s rezervou dosáhl. Samotné utahování konstrukce šroubů probíhalo diagonálně s doplněným měřením pomocí vodováhy, abych dosáhl co nejlepší stability konstrukce („neviklala se“ díky rozdílnému smontování). Finální konstrukci, jsem ještě ověřil proti zkrutu vzniklé při montáži pomocí diagonálních rozměrů horní a dolní podstavy (tyto rozměry totiž musely být stejné).

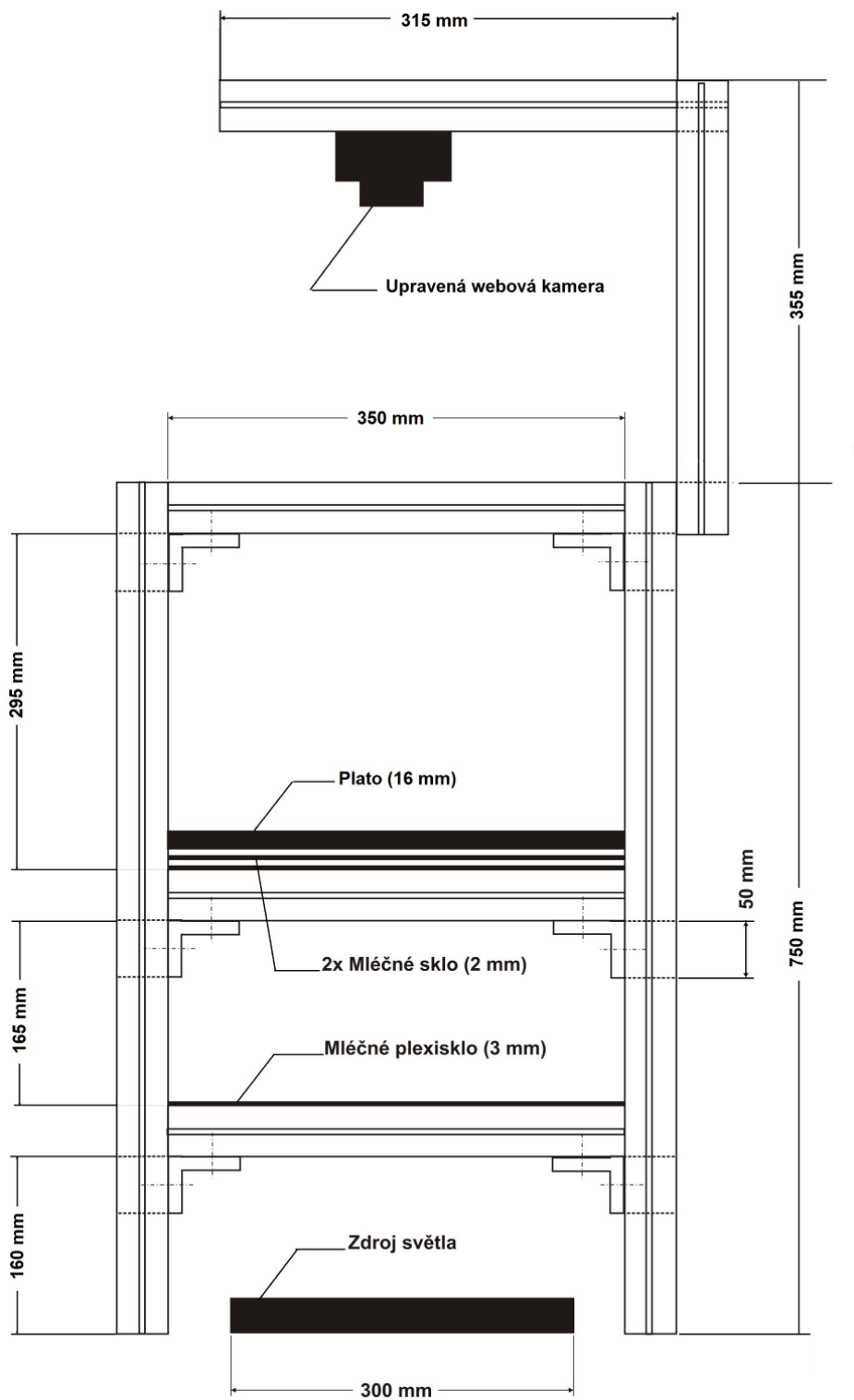
Nátěr konstrukčních tyčí bylo po smontování celé finální konstrukce potřeba ještě doopravit, došlo totiž k jeho poškrábání při manipulaci (samotný nátěr na tyčích s hliníku příliš nedržel, i když byly před nanesením nátěru několikrát odmaštěny pomocí technického acetonu). Žlábků uvnitř konstrukčních tyčí (uzpůsobené pro montáž solárních panelů) jsem v první fázi testů používal jako vodiče a držáky síťových a napájecích kabelů tudíž bylo potřeba celou konstrukci dodatečně uzemnit, aby nedošlo k případnému mechanickému poškození kabelu a zranění obsluhy zařízení (průnik 230 V na kostru konstrukce). Velké kruhové světlo těchto žlábků nevyužívá a jeho přívodní šňůra je vedena mimo konstrukci.

Na obr. č. 7 je znázorněn náčrtek modelu s výslednými technickými rozměry a uspořádáním celé konstrukce. Samotný profil hotových konstrukčních tyčí je zobrazen na obr. č. 1 a díky unifikovaným rozměrům (hrana 45 mm) nebylo nutné konstrukci těmito rozměry přizpůsobovat, proto je již náčrtek modelu na těchto parametrech tyčí založen.

Konstrukční tyče na stavbu poskytla firma zabývající se stavbou a montáží fotovoltaických setů pro koncové uživatele, tudíž pořizovací náklady na stavbu konstrukce přístroje pro sledování a vyhodnocování pohybové aktivity hmyzu nenarostli o tuto cenu. Profily, respektive hrany tyčí byli nařezány profesionálním laserem, tudíž jejich hrany dosedaly, takřka dokonale, což se nedalo říci o klínech, které jsem použil pro fixaci těchto tyčí k sobě samotným. Mírné přesahy klínů rozhazovaly geometrii konstrukce a byla nutná jejich úprava pomocí brusky.

Z náčrtku na obr. č. 7 je také očividné připevnění webové kamery ke konstrukci. To je realizováno pomocí vytvořeného ramene (tvoří jej 2 konstrukční tyče vhodné délky) jehož šrouby jsou záměrně lehce povoleny, lze s ním poměrně lehce manipulovat a umožňuje tak manuální vertikální i horizontální pohyb webové kamery. Vystředění kamery na záběr probíhajícího děje je potřeba provést manuálně. K vystředění jsem používal v záběru okraje konstrukčních tyčí prostředního patra samotné konstrukce přístroje (byť nejde o sofistikovanou metodu, je tato metoda zřejmě nejrychlejší a mírné nedostatky je možné doladit pohybem samotného plata s dějem – je mu nechána vůle cca 5 mm). Samotná vzdálenost webové kamery a samotného snímaného děje (plato s mouchami) činí v závislosti na nastavení cca 50 cm, tato vzdálenost je ovlivněna použitým objektivem a jeho snímacími schopnostmi (úhel záběru objektivu). V případě

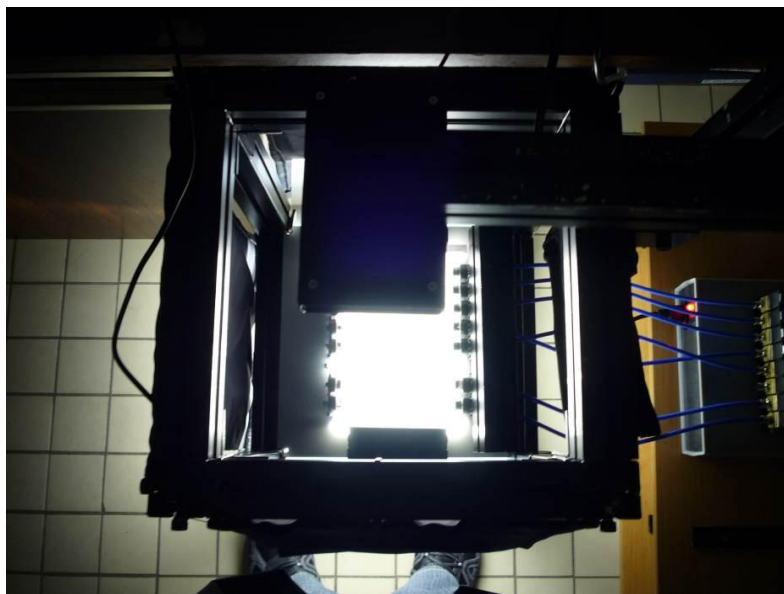
že by byl použit objektiv s velkou ohniskovou vzdáleností, bylo by potřeba kameru dostat do poměrně vysoké výšky, což lze zajistit prodloužením konstrukčního ramene dalšími tyčemi spojenými spojkami.



Obr. č. 7 Původní technický náčrtek modelu konstrukce přístroje pro sledování a vyhodnocování pohybové aktivity hmyzu.

Na technickém náčrtku jsou také znázorněny plánované skleněné desky, které zde plní funkci homogenizace světla čili rovnoměrného rozptylu světla do co možná největší plochy. Snímaná plocha se pak jeví pro video záznam, jako kompaktně podsvícená což usnadňuje analýzu po softwarové stránce.

### 3.2.5 Finální konstrukce



Obr. č. 8 Horní pohled na finální konstrukci přístroje pro sledování a vyhodnocování pohybové aktivity hmyzu přímo při provozu (v záběru patrná webová kamera)



Obr. č. 9 Boční pohled na finální konstrukci přístroje pro sledování a vyhodnocování pohybové aktivity hmyzu (vpravo v záběru patrný také dávkovač vzduchu)

### 3.3 Volba záznamového zařízení

Výběr obrazového záznamového zařízení byl zpočátku omezen následujícími kritérii:

- Pořizovací cena do 2000kč
- Minimální snímková frekvence 15 FPS při rozlišení 800x600
- Fyzické rozlišení čipu minimálně 2 Mpx
- Manuální (HW, SW) nastavení parametrů (uzávěrka, jas, barvy atd.)
- Zařízení bude možné snadno upravit po HW stránce
- Záznam v reálném čase bez dodatečných převodů záznamu
- Snadná manipulace s pořízeným audio-video-materiálem
- Přenos dat do PC v reálném čase (USB, FireWire, Wi-Fi)

Předně kritéria eliminovala veškerá analogová zařízení, která by dodatečně vyžadovala čas při převodu pořízeného záznamu do digitální formy vhodné pro počítač a vyhodnocovací SW, nehledě na nutnost vlastnit tzv. „grabovací zařízení“, které by umožnilo výše popsaný převod a dále pak nutné zaškolení obsluhy konstruovaného měřicího přístroje pro analýzu pohybové aktivity hmyzu. Dále byly z výběru eliminovány digitální fotoaparáty a videokamery, které v této cenové hladině nemají vyměnitelný objektiv, neumožňují snadnou HW úpravu, poskytují nízká rozlišení a kvalitu zaznamenaného videa.

Volba padla na oblast cenově dostupných webových kamer, které splňují veškerá kritéria výběru a jejich výběr na trhu je dostatečně široký. Možnost HW úpravy, přímé spojení s PC, přívětivá cena a všestranné nasazení jsou jedny z hlavních důvodů, proč jsou velmi často nasazovány v amatérských a poloprofesionálních použitích jako neoptimálnější a zároveň dostatečné zařízení s optimálním poměrem cena/výkon. Dalším kladem použití je kompatibilita webových kamer napříč operačními systémy (Windows, Linux), je možné je tedy použít a přistupovat k nim přímo ze specializovaných a vyvíjených SW a získat v „reálném čase“ požadovaná data v rozličných formátech bez nutnosti dodatečné konverze či přenášení dat z jiných médií (paměťová karta, flash disk, DVD) či dodatečného „grabování“ dat.

Nejlepší poměr cena/výkon v oblasti webových kamer nabízela v době realizace přístroje pro sledování pohybové aktivity hmyzu webová kamera 9000 Pro od firmy Logitech, na jejímž základě jsou konstruovány nové generace HD kamer s obměňovanými technologiemi. Její technické parametry [1] jsou následující:

- Carl Zeiss® optika s automatickým ostřením
- Fyzický 2 Mpx HD čip (nativní rozlišení 1600x1200 pro fotografie)
- HD video (až 1600 X 1200 – interpolované rozlišení)
- 8 Mpx fotografie (interpolovány z fyzických 2 Mpx čipu)
- Mikrofon s technologií Logitech® RightSound™

- 30 FPS ve zvolených režimech (až 1280x720p)
- Vysokorychlostní USB 2.0 připojení
- Ovladač s grafickým rozhraním (možnost manuálního nastavení parametrů kamery)
- Univerzální uchycení pro CRT a LCD monitory
- Digitální zoom

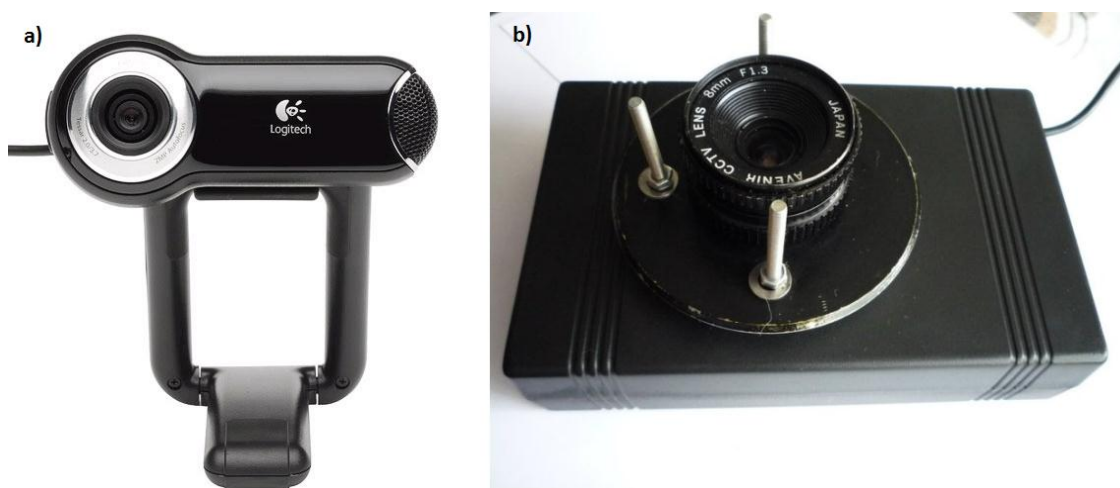
### 3.3.1 Detailní hlediska výběru

#### A) Uchycení a konstrukce

Externí webové kamery je možné umístit před snímání obraz různými způsoby. Většina prodávaných externích webových kamer je připravena pro spolupráci s PC (vyjímaje LAN kamery, které mívají obvykle jiný systém uchycení a typ montáže), proto jsou vybaveny konstrukcí (nejčastěji stojanem, čelistmi či klipsem), která umožňuje umístění webové kamery přímo na stůl či monitor (CRT), či přichycením na tenkou hranu LCD obrazovky.

Většina webových kamer na trhu je opatřena plastovou konstrukcí šasi, která chrání DPS se samotným čipem a dalšími komponenty (mikrofon, led indikace zapnutí) před mechanickým a jiným poškozením.

Výběr uchycení kamery byl v mém případě nepodstatný, kameru jsem totiž byl nucen mechanicky upravit pro potřeby uchycení na masivní stojan a dále bylo potřeba zajistit vyměnitelnost objektivů, tyto úpravy pak umožní variabilnější využití webové kamery pro další použití (mikroskop atd.) jak je vidět při konfrontaci obr. č. 10a) a finální verzi HW úpravy na obr. č. 10b).



Obr. č. 10 a) Standartní konstrukce Logitech 9000 Pro převzato a upraveno z [2], b) Upravená konstrukce Logitech 9000 pro (vyměnitelný objektiv, systém uchycení, šasi)

Úpravou konstrukce uchycení a zvětšením šasi bylo zajištěno lepší chlazení čipu a DPS, odstraněním napevno zafixovaného objektivu výrobce a úpravou konstrukce upevnění objektivu na šroubový závit (40mm) byla zajištěna možnost montáže a výměny průmyslových objektivů s různými parametry (světlost, filtry, ostření).

## B) Snímková frekvence a možnosti nastavení GUI ovladače

Frekvence snímkování je jinými slovy schopnost kamery dosahovat plynulého obrazu, čili počet nasnímaných obrazů reality za jednotku sekundy. Snímková frekvence se nejčastěji označuje FPS (Frame Per Second).

Pro dosažení plynulého obrazu pro lidské oko, tak aby oko nerozeznávalo jednotlivé obrazy videa, by měla být kamera schopna zaznamenávat video se snímkovou frekvencí nejméně 25 snímků za sekundu (odpovídá 25 FPS). Je-li obraz plynulejší, tzn.  $FPS > 25$ , lidské oko ho vnímá stále stejně plynule (jde o subjektivní vjem). Pokud je snímková frekvence nižší než 25 FPS, obraz se jeví jako „rthany“.

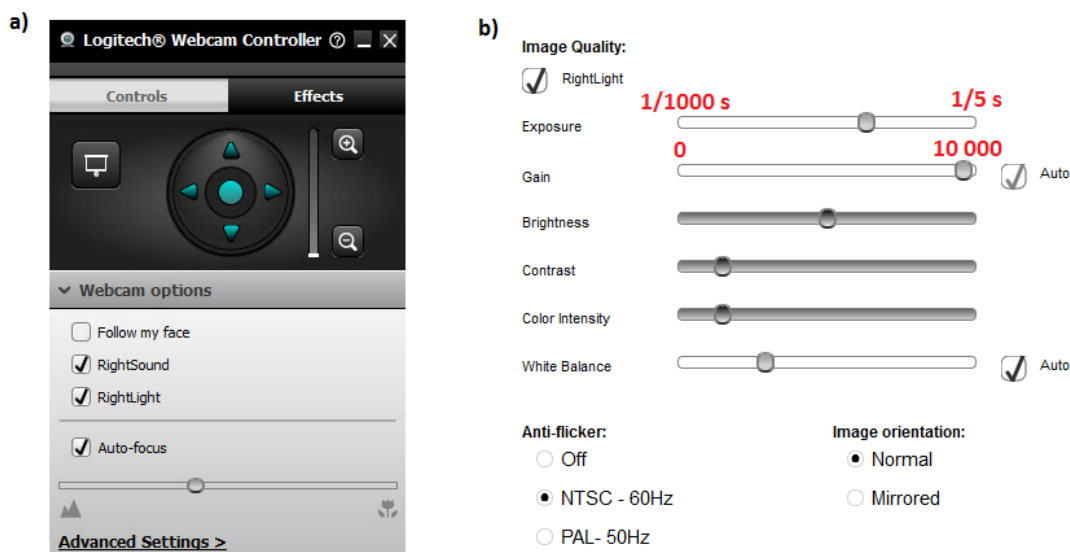
Snímkovou frekvenci nejen webové kamery ale i ostatních obrazově záznamových zařízení ovlivňuje řada faktorů, zejména technologie výroby čipu (CMOS, CCD), dále množství světla dopadajícího na čip – výrobci jej totiž záměrně „spojují“ s řízením uzávěrky kamery a v neposlední řadě i požadované výstupní rozlišení obrazového materiálu). Obecně jsou na světlo lépe citlivé CCD čipy (méně šumí při nízkých citlivostech ISO) [3].

Snímková frekvence je jeden z hlavních údajů uváděný výrobcem na obalu od webové kamery. Výrobci levných kamery s rozpětím velmi nízkých rozlišení (cca do 1,3 Mpx) uvádějí jako maximální snímkovou frekvenci 30 FPS při rozlišení uváděném obvykle 640x480 px. Této snímkové frekvence však není možné dosáhnout bez ideálních světelných podmínek, protože uzávěrka webkamery (pohyblivá clona, která umožňuje přístup a krytí čipu před světlem) je elektronicky řízena samotnou kamerou a dále pak ovladačem OS pro dosažení konstantní snímkové frekvence. Vhodnými úpravami SW třetích stran či přímo ovladače (pokud výrobce uzpůsobil GUI ovladače tomuto nastavení) je možné činnost uzávěrky korigovat manuálně (nastavit pevnou hodnotu uzávěrky potažmo FPS), mnohdy však na úkor světelnosti obrazu jak je vidět na obr. č. 2, který ukazuje možnosti manuálního nastavení u vybrané kamery Logitech 9000 Pro:

Na obr. č. 11b) je defaultní nastavení od výrobce. Pro manuální nastavení webové kamery je potřeba odškrtnout položku „RightLight“ (technologie pro zajištění kvalitního obrazu za jakýchkoliv světelných podmínek automatickým přizpůsobením položek „exposure“ a „gain“, čili expozice a nastavení zesílení gain) a položku „Auto“ u zesílení „Gain“, poté je již možné manuálně nastavovat expozici (potažmo uzávěrku webkamery) a zesílení tzv. gain, s případnou korekcí v podobě světlosti obrazu („Brightness“), kontrastu obrazu („Contrast“), barevné intenzity („Color Intensity“) a vyvážení bílé („White Balance).

Obr. č. 11a) zobrazuje možnosti nastavení zoomu (v případě 9000 Pro pouze digitálního), dále pak možností zapnutí technologií „RightLight“ (popsána výše) a „RightSound“ (obdoba vylepšení obrazu RightLight avšak týkající se zvuku z integrovaného mikrofónu ve webové kameře). Automatické ostření „Auto-focus“ lze vypnout a použít manuálního nastavení (ostření je realizováno elektronicky miniaturním motorkem pohybujícím s čočkou objektivu). Ostření je po úpravě webové kamery

nefunkční, celý objektiv je totiž včetně elektronického ostření a zoomu odmontován (ostření je nutné provádět manuálně přímo na průmyslovém objektivu).



Obr. č. 11 a) Nastavení zoomu a ostření webové kamery Logitech 9000 Pro, převzato a upraveno z [4], b) Detailní nastavení parametrů webové kamery Logitech 9000 Pro, převzato a upraveno [4]

Dle [5] je nutné nastavit „Anti-flicker“ (eliminace blikání a rušení v obraze) na „Off“, zapnutím této eliminace na 50Hz je omezena expozice na 1/120 s, při nastavení na 60Hz je pak expozice omezena na minimální hodnotu 1/100 s. Eliminace rušení a „blikání“ v obraze je vhodné vypínat pouze v případech, kdy nedochází ke snímání obrazu pod zářivkovým světlem (síťové napájecí napětí zářivky v ČR 50Hz interferuje s obrazem LCD či CRT monitoru 50-60 Hz), paradoxně jde pouze o vizuální interferenci, která se neprojevuje u nahraného záznamu, pouze u tzv. „live“ režimu (náhledu obrazu webové kamery). Interference v obraze se projevuje jako několik pravidelných tmavých pruhů postupně a harmonicky se pohybujících po obraze náhledu webové kamery, změnou uzávěrky (buď přes nastavení dodávaného SW k nastavení ovladače webové kamery) nebo pomocí manuálního nastavení clony objektivu je možné dosáhnout optimálního stavu bez interference.

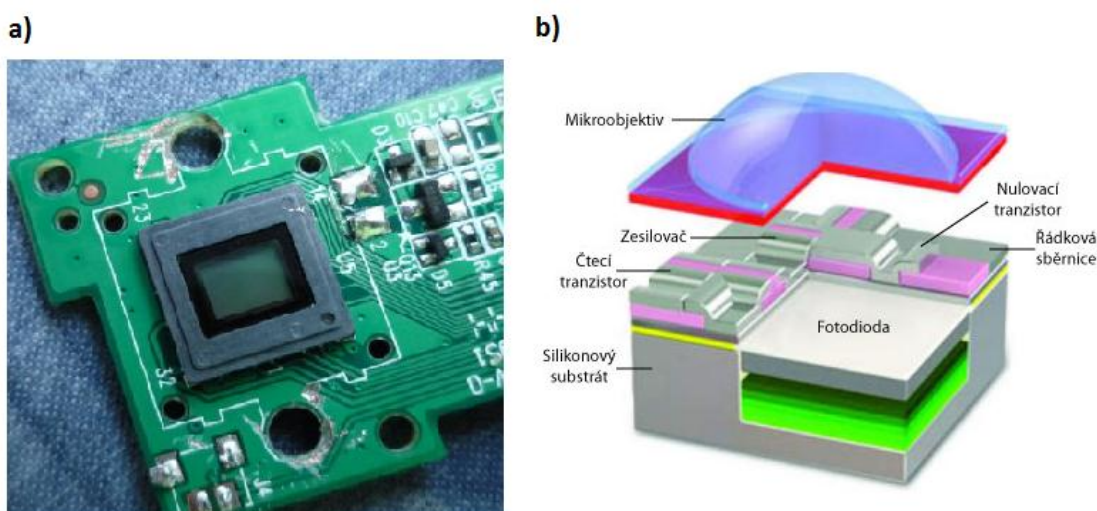
### C) Druh použitého čipu

Webové kamery jsou dnes osazovány nejčastěji jedním ze dvou druhů čipů odlišných konstrukcí i výrobních technologií.

Starší technologie, ale stále používaná (zejména v dnešních kamerách, integrovaných fotoaparátech mobilních telefonů), vede k výrobě CMOS čipů. Zjednodušeně lze dle [6] poukázat na velmi nízkou spotřebu (pro čtení dat z čipu je potřeba pouze jedné napěťové úrovně a navíc má každý bod čipu citlivý na světlo vlastní logický obvod s výstupem), takřka okamžité čtení dat z čipu (není nutný posun



a odečítání náboje jako u CCD, data se čtou najednou ze všech výstupů ať už řádku či sloupce světločivých bodů čipu) s čímž souvisí i možné kratší časy expozice čipu (doba, po kterou na čip dopadá světlo potažmo „snímaný obraz“) a tím i kratší doby uzávěrky (pohyblivá clona která zabraňuje přístupu světla na čip), levnější výroba a snadná miniaturizace (výrobní technologie je podobná výrobě procesorů). Avšak díky potřebě výstupu pro každý jednotlivý světločivý výstup čipu (stovky milióny obvodů) bylo velmi obtížné dosáhnout optimálního poměru výrobní cena/rozlišení čipu, což byl také jeden z důvodů, proč byly tyto druhy čipů dříve vyráběny spíše s nižšími fyzickými rozlišeními a používány jako srdce levnějších a méně kvalitních obrazově-záznamových zařízení, byť s lepšími šumovými parametry (čip ovlivňuje méně vyprodukovaného tepla) než dražší a modernější CCD čipy. Dnes jsou však díky velkému pokroku v miniaturizaci výrobci schopni vyrobit i čipy technologií CMOS s rozlišením více jak 100 Mpx [7].



Obr. č. 12 a) Detail CMOS čipu webové kamery Logitech 9000 Pro, b) Schématické znázornění jedné buňky CMOS senzoru, převzato a upraveno z [8]

Druhá a novější technologie, nyní na ústupu oproti CMOS technologii, jejímž produktem je CCD čip (zkratka názvu Charged-Coupled Device) je oproti CMOS čipům jednodušší na výrobu, avšak cenově nákladnější. Čipy sice dosahují extrémních rozlišení, ale na úkor doby uzávěrky, která je ve srovnání s CMOS čipy delší (data, respektive vázané náboje jednotlivých bodů citlivých na světlo se z čipu načítají postupně po sloupcích či řádcích, je však nutné zajistit posun jednotlivých nábojů v sloupci či řádku). Větší spotřeba čipu způsobuje větší produkované teplo oproti CMOS a tím i horší šumové vlastnosti CCD, které je však možné eliminovat chlazením čipu (díky použití křemíku je CCD čip relativně teplotně stálý [6]). Obecně je ale CCD čip výhodnější použít v oblasti s nižšími světelnými podmínkami (CCD čip je mnohonásobně citlivější na světlo než CMOS [9]), to je také jeden z hlavních důvodů proč jsou tyto druhy čipů používány mnoha astronomy [3].

Oba druhy čipů produkují surová analogová data, je potřeba je proto ještě následně zpracovat do digitální formy pomocí A/D převodníků. Digitální forma obrazu je výhodná zejména pro následné matematické zpracování úpravy obrazu (odstranění

vad optického systému – zejména objektivu, jako je např. barevná vada, soudkovitost, poduškovitost, dále pak konverze dat do vhodného obrazového formátu, jako je např. RAW, JPEG atd.)

Zvolená webová kamera Logitech 9000 Pro je založena na čipu vyrobeném technologií CMOS jak je vidět na obr. č. 12a).

#### **D) Rozlišení a požadované rozhraní pro připojení**

U webových kamer je pod pojmem rozlišením míněn počet obrazových bodů, tj. pixelů (pixel = nejmenší jednotka bitmapového obrazu), ze kterých se skládá výsledný obrázek (video) v nejlepší možné kvalitě (uváděno obvykle výrobcem na obalu od webové kamery). Rozlišení webové kamery je striktně závislé na HW řešení čipu webové kamery a samotných schopnostech čipu. Výrobce může uvádět rozlišení ve dvou tvarech, buď v „megapixelech“, což je počet pixelů v obrazu ( $1\text{Mpx} = 1 \cdot 10^6 \text{px}$ , jak uvádí [1]), nebo přímo uvedením horizontálního a vertikálního rozlišení, čili počtu obrazových bodů v horizontální a vertikální rovině (např.  $800 \times 600 \text{px}$ ). Platí přímá úměra mezi rozlišením a kvalitou obrazu, čím větší rozlišení tím kvalitnější pořízený obraz. Dnešní moderní videokamery, digitální fotoaparáty ale už i webové kamery mají standardní rozlišení převyšující vysoko 2Mpx, což odpovídá vertikálně a horizontálně rozlišením větším než  $1600 \times 1200 \text{px}$ .

Rozlišení čipu uvedené v „megapixelech“ ještě není zárukou, že výsledná výstupní obrazová data budou v tomto rozlišení z čipu dostupná. Někteří výrobci totiž používají minimální část plochy čipu (do 5 až 10%) pro měření tzv. „nultých hladin obrazu“, tzn. jasu, barev apod. Toto vyhrazení plochy respektive části světločivých bodů způsobí, že i když je čip schopen fyzicky dosahovat vyšších rozlišení (omezených jeho fyzickou konstrukcí), tak výstupní data jsou v rozlišeních nižších, případně vyšších (tzv. interpolovaných či přepočtených).

Jak je vidět z tabulky 1, datový tok potřebný pro přenos záznamu o rozlišení  $800 \times 600$  při 24bitové barevné hloubce s barevným modelem YUV 4:2:2 a 15 FPS je pro rozhraní USB 1.0 příliš náročný (172,8 Mb/s), přičemž maximální datová propustnost USB 1.0 činí 12 Mb/s, avšak USB 2.0 (s maximální propustností 480 Mb/s) je již pro záznam s výše uvedenými parametry více než dostatečný (maximální využití širší pásma při zmíněném přenosu po USB 2.0 činí 72 %, což dává rezervu i při použití s pomalejšími USB řadiči). Schopnosti webové kamery 9000 Pro (zejména fyzické rozlišení CMOS čipu 2 Mpx) a její připojení přes USB 2.0 umožňují záznam i v pseudo-HD rozlišení  $1280 \times 720$  při 15 FPS, 24 bitové barevné hloubce a barevném modelu YUV 4:2:2, což zajistí dostatečnou robustnost použití webové kamery i k jiným účelům.

Tabulka 1 Srovnání datových toků potřebných pro přenos videa pro USB 1.0 a USB 2.0

Rozlišení (px)	Rozlišení (Mpx)	Barevná hloubka (bit)	Barevný model	FPS	Datový tok (Mb/s)	Využití USB 1.0 (%)	Využití USB 2.0 (%)
160x120	0,019	24	YUV 4:2:2	15	6,91	57,6	1,4
			YUV 4:2:3	30	13,82	115,2	2,9
320x240	0,076	24	YUV 4:2:4	15	27,65	230,4	5,8
			YUV 4:2:5	30	55,30	460,8	11,5
640x480	0,307	24	YUV 4:2:6	15	110,59	921,6	23,0
			YUV 4:2:7	30	221,18	1843,2	46,1
800x600	0,480	24	YUV 4:2:8	15	172,80	1440,0	36,0
			YUV 4:2:9	30	345,60	2880,0	72,0
1024x768	0,786	24	YUV 4:2:10	15	283,12	2359,3	59,0
			YUV 4:2:11	30	566,23	4718,6	118,0
1280x720	0,921	24	YUV 4:2:12	15	331,78	2764,8	69,1
			YUV 4:2:13	30	663,55	5529,6	138,2
1600x1200	1,920	24	YUV 4:2:14	15	691,20	5760,0	144,0
			YUV 4:2:15	30	1380,00	11500,0	287,5

(\* Datový tok byl kalkulován pomocí software [10], vyšší rozlišení je webová kamera schopna poskytnout pouze v interpolované podobě a díky vysokému datovému toku již do tabulky zahrnuty nejsou).

### E) Interpolace

Důležité je také fyzické rozlišení čipu a rozlišení pořizovaného záznamu (obraz, video). Výrobci často na obal od zařízení uvádějí tzv. interpolované rozlišení, tedy nikoliv rozlišení, které je fyzicky čip schopen snímat, ale upravené (přepočtené) na vyšší hodnoty (zejména u statických obrazů – fotografií). Proces interpolace dle [11], je metoda, která vylepšuje a zvyšuje rozlišení obrazu vložení „uměle dopočtených“ pixelů mezi původní, nejčastěji sousední pixely obrazu. Metod interpolace [12] (převzorkování) existuje celá řada, nejjednodušším způsobem je tzv. lineární či aproximační interpolace, ta spočívá v tom, že se hodnota barevné složky interpolovaného pixelu, spočítá jako aritmetický průměr dvou sousedních hodnot pixelů původního obrazu, viz následující rovnice:

$$A_{\text{ipx}} = A_{(i-1)} + \left( |A_{(i+1)} - A_{(i-1)}| / 2 \right), \quad (1)$$

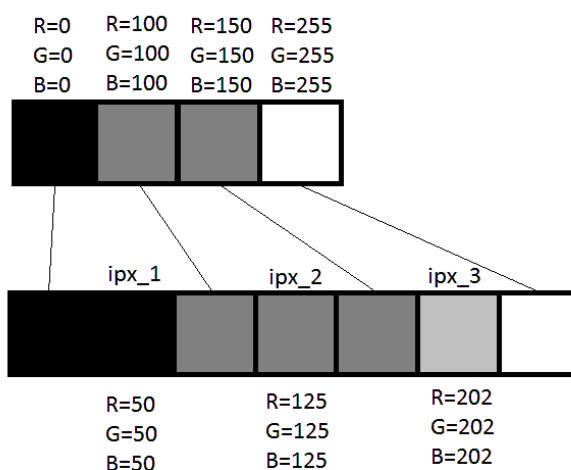
$$R_{\text{ipx}} = R_{(i-1)} + \left( |R_{(i+1)} - R_{(i-1)}| / 2 \right), \quad (2)$$

$$G_{\text{ipx}} = G_{(i-1)} + \left( |G_{(i+1)} - G_{(i-1)}| / 2 \right), \quad (3)$$

$$B_{\text{ipx}} = B_{(i-1)} + \left( |B_{(i+1)} - B_{(i-1)}| / 2 \right), \quad (4)$$

Lineární interpolace je vhodná pro zvětšování obrazu do maximální velikosti 100% oproti původnímu obrazu, poté je již velmi patrné „zrnění“ obrazu, některé dnešní levné webové kamery s malým rozlišením čipu (obvykle kolem 1 Mpx) a fotoaparáty ji dodnes používají. Moderní a sofistikovanější webové kamery (např. i Logitech 9000 Pro) a další zařízení používají také tzv. hybridní aproximační bikubickou interpolaci, ta spočívá v proložení čtyř sousedních bodů původního obrazu polynomem třetího stupně (kubickou křivkou), rovnice křivky pak umožní dopočítat libovolný počet bodů uvnitř oněch čtyř bodů, z toho vyplývá dosažení lepší věrnosti zvětšeného obrazu i v případě většího zvětšení než u lineární interpolace jak uvádí [12].

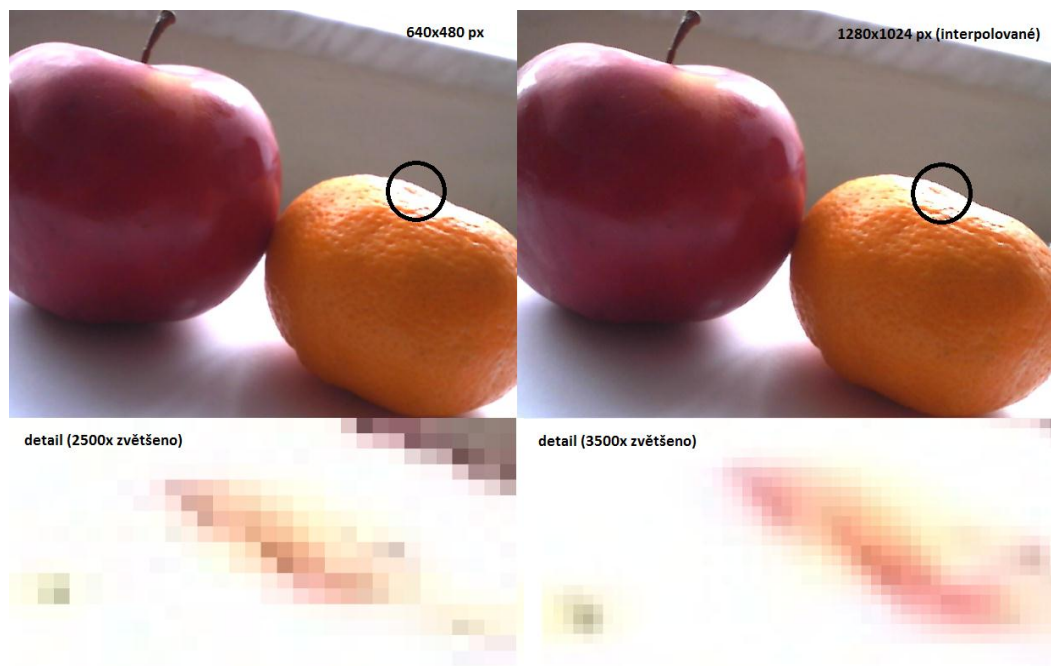
Na obr. č. 13 je znázorněno, jak k lineární interpolaci dochází. Jde o rozšíření ze čtyř na sedm pixelů (čili takřka dvojnásobné rozlišení oproti původnímu obrazu). Původní pixely zaujmou nová místa (krom prvního) a mezi nimi dochází ke kalkulaci barevných složek mezi-pixelů pomocí výše uvedených rovnic (2) až (4) v případě 24bitového RGB systému. „R“ znázorňuje hodnotu červené složky, „G“ je zelené složky a „B“ je hodnotou modré složky pixelu.



Obr. č. 13 Schématické znázornění lineární interpolace obrazu (přepočítání 4px na 7px)

Obr. č. 14 poté ukazuje konkrétní případ interpolace. Na snímku vlevo je zobrazen původní obraz zachycený v rozlišení 640x480 vybranou webovou kamerou Logitech 9000 Pro s náhledem hrany, vpravo je pak tentýž snímek, avšak s tím rozdílem, že zobrazení odpovídá interpolovanému snímku vlevo s rozlišením 1280x1024. Pravý detail hrany ukazuje zcela zřejmou funkci interpolace.

Technicky by interpolace obrazové analýze neměla způsobovat problémy, avšak z hlediska reálného měření nastává problém. Pixely které tvoří objekt jsou totiž uměle dopočítávány, což by v případě pixelů uvnitř objektu příliš nevadilo, ale po okrajích objektu (mezi pixely pozadí a objektu) jsou virtuálně taktéž dopočítány mezipixely, které by ovlivňovaly korektnost analýzy, proto není příliš vhodné používat interpolovaný záznam (video, obraz) pro následnou obrazovou analýzu.

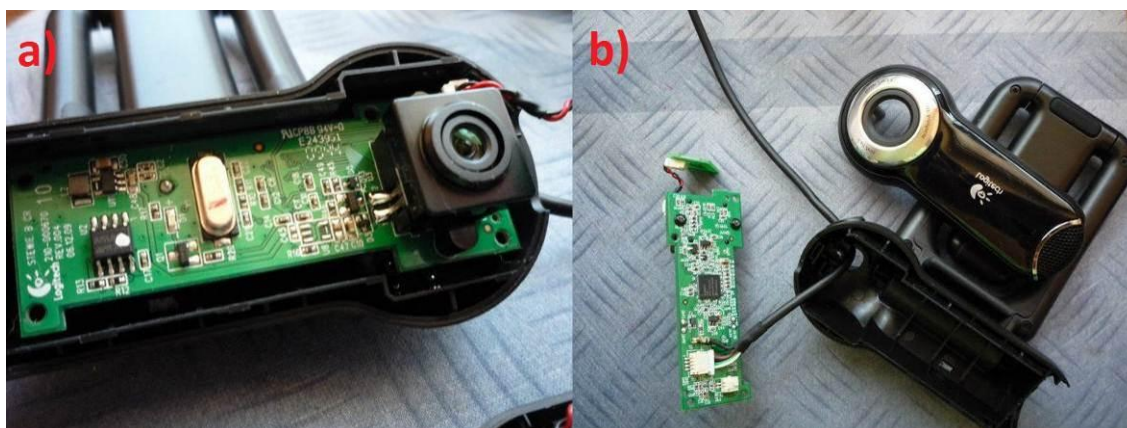


Obr. č. 14 Příklad interpolovaného snímku pořízeného webovou kamerou Logitech 9000 Pro

### 3.3.2 Hardwarová úprava webové kamery

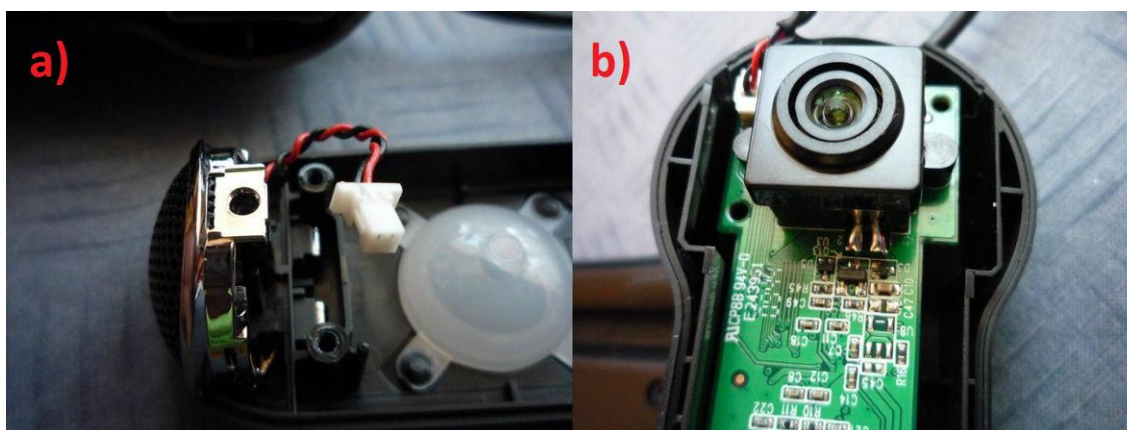
Výrobce standardně dodává k prodeji webovou kameru Logitech 9000 Pro složenou ze dvou hlavních částí, ze samotného těla (lesklého plastového krytu DPS včetně kruhového indikátoru oranžové LED diody, vodítka objektivu a mřížky pro integrovaný mikrofón) a úchytu pro LCD či CRT monitor. Ze samotného těla kamery vede 1,8 m dlouhý kabel s USB koncovkou typu A (samice).

Prvním krokem úpravy bylo odstranění krytu těla webové kamery. Povoláním dvou šroubků v zadní části těla webové kamery bylo možné rozebrat plastové tělo, jak je to zobrazeno na obr. č. 15b). Klips či úchytka pro monitor nebylo nutné rozmontovávat, jelikož bylo potřeba získat přístup pouze k DPS s čipem a objektivu, proto jsem nechal zadní část těla kamery i klips (úchytka) smontované do jedné kompaktní části pro případné zpětného navrácení DPS do původního originálního krytu. Z obr. č. 15a) je vidět, že DPS je přichycena čtveřicí malých šroubků umístěným na okrajích DPS, po vyšroubování těchto šroubků je možné DPS vyjmout jak je vyobrazeno na obr. č. 15b) - obrázek znázorňuje již odpojený mikrofón (volný bílý konektor v dolní části DPS). Na tomto obrázku je také znázorněna menší DPS připojená dvoužilovým (červeno-černým kabelem) k hlavní DPS s čipem. Skrývá na sobě mikropínač a obvody zajišťující korekci zákmitů (tento spínač slouží pro rychlé pořízení snímků pomocí webové kamery).



Obr. č. 15 a) Odstraněný přední kryt těla webové kamery Logitech 9000 Pro (odmontovány šroubky držící DPS po jejich okrajích); b) Kompletně rozebrané tělo webové kamery Logitech 9000 Pro

Na obr. č. 16a) je zobrazeno propojení hlavní DPS a integrovaného mikrofону (integrován v plastovém těle webové kamery, je však vyjímatelný). Připojení mikrofónu tvoří dvojice vodičů s rozlišenou polaritou (červený a černý) zakončeného unifikovaným konektorem, bohužel se mi nepodařilo dohledat, o jaký typ mikrofónu se jedná. K mým účelům, však důležitý nebyl (primárně je nutné pořizovat obrazový záznam, nikoliv zvukový) a díky nesnadné implementaci mikrofónu (jde zřejmě o směrový mikrofón) do nové konstrukce. Na tomto obrázku je také vidět bílá plastová záslepka (z opačné strany těla, na vnější části, pokrývá přesně plochu firemního loga Logitech na lesklém plastu, viz obr. č. 15b). Tato poloprůhledná plastová záslepka slouží pro dokonalé rozprostření (homogenizaci) světla z indikační SMD LED diody, která informuje o tom, zdali je webová kamera zapnuta či zrovna pořizuje záznam (tato indikace lze vypnout a nastavit v GUI ovladači dodávaném k webové kameře). Při indikaci je díky této záslepce na vnějším těle kamery rozsvícen oranžový kruh kolem loga výrobce.



Obr. č. 16 a) Mikrofón a jeho připojení k hlavní DPS webové kamery Logitech 9000 Pro; b) Připojení a řízení objektivu včetně elektronického ostření k hlavní DPS webové kamery Logitech 9000 Pro

Na obr. č. 16b) je vyobrazeno připojení objektivu respektive automatického (elektronicky řízeného) ostření k hlavní desce webové kamery. Tvoří jej 2 napájené

vodiče vedoucí k objektivu (+5 V a GND). Napájení ostření je realizováno pomocí USB (+5 V a až 500 mA).

Na obr. č. 17a) je možné pozorovat horní stranu hlavní DPS s řídicím čipem. Mnou pořízená webová kamera Logitech 9000 Pro obsahuje DPS revize verze 004 výrobního data 6. 12. 2009, tato revize používá krom standartních datových (DATA\_IN = zelený vodič, DATA\_OUT = bílý vodič) a napájecích vodičů (+5 V = červený vodič, GND = černý vodič) USB sběrnice navíc ještě jeden vodič, na obr. č. 17a) s tlustou černou izolací, tzv. druhý zemnicí respektive stínící vodič, který je připájen na pinu procházející celou tloušťkou desky a zajišťuje zemnicí vodič dostupný pro horní a dolní stranu DPS, která je tudíž oboustranná. Jelikož byla samice připojovacího konektoru USB na konci kabelu příliš široká a nešla protáhnout plastovým otvorem v těle webové kamery (obr. č. 15b) a samotné tělo nešlo již dále bez destrukce demontovat, bylo nutné odpojit konektor kabelu na opačném konci, tedy přímo z DPS, to však vyžadovalo odpájení stínícího vodiče.

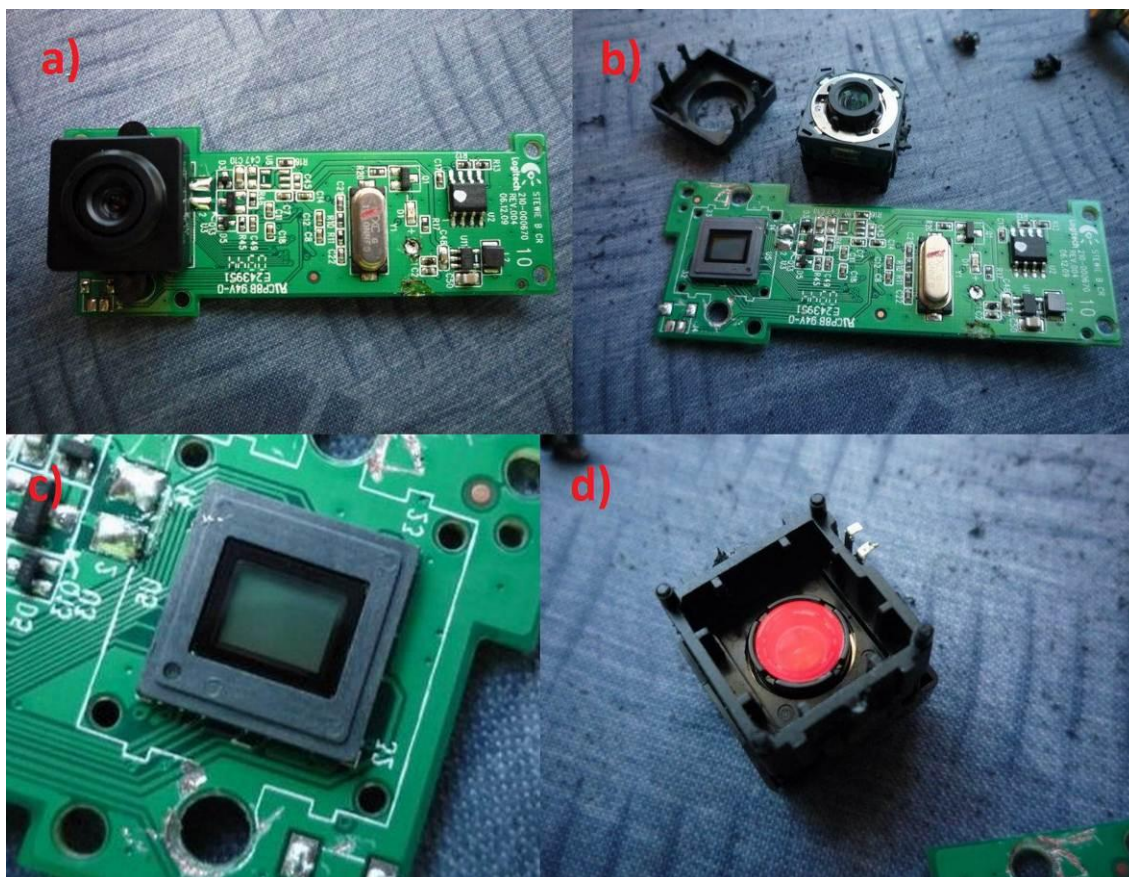


Obr. č. 17 a) Náhled DPS webové kamery Logitech 9000 Pro s připojeným hlavním datovým kabelem USB; b) Detail DPS webové kamery Logitech 9000 Pro s odpájeným stínícím vodičem (spodní strana DPS)

Na dalším obr. č. 18a) je již vidět samotná deska DPS s objektivem. Za kvalitu objektivu ručí švýcarská firma Carl Zeiss, čili jde o poměrně kvalitní doplněk CMOS senzoru webové kamery (na trhu je velmi málo kamer s tímto prestižním certifikátem). Objektiv byl však také nepotřebný a bylo nutné ho demontovat, to však nebylo tak snadné, objektiv (včetně elektronického ostření) držely, jak si můžete na obrázku po stranách dva šrouby (na obr. č. 18a) jsou vidět pouze výlisky pro závity šroubů, je vyfocena totiž spodní strana DPS přičemž hlavičky šroubů se nacházejí na horní straně DPS) a dva připájené vodiče ostřicího systému přímo na základní desku. Odpájení dvou vodičů šlo hladce, avšak zdánlivě lehké odšroubování dvou šroubků po stranách objektivu nešlo tak hladce, šrouby byly zalité lepidlem jak do svých vlastních závitů tak i do DPS (firma Logitech mi přes svůj support email sdělila, že důvod takovéto fixace, u DPS s revizním číslem 004, je následující, jednak pevně a přesně uchytit objektiv a dále pak znemožnit snadnou demontáž objektivu, webová kamera Logitech 9000 Pro byla totiž velmi často využívána i pro jiné než pro své primární určení zejména astronomy). To způsobilo, že nešly povolit ani hrubou silou. Uchýlil jsem se proto

k nucenému odvrtání šroubků po obou stranách plastového krytu objektivu včetně odstranění lepidla.

Obr. č. 18b) zobrazuje již demontovaný objektiv s obnaženým mikromotorem pro ostření a samotnou optikou objektivu. Objektiv se chlubí pozorovacími úhly v rozmezí  $75^\circ$  od normály [1]. Odkrytý CMOS čip zobrazený na obr. č. 18c) o velikosti 4,4 mm x 3,3 mm, diagonálně 5,5 mm, jehož hodnota poměru šířky k výšce činí standardních 4:3 (širokoúhlé zobrazení je umožněno nepoužíváním části plochy čipu, poměr šířky k výšce je pak uplatněn jako 16:9, ve prospěch šířky čipu). Velikost 1 px na čipu odpovídá cca 3,5  $\mu\text{m}$  [13].



Obr. č. 18 a) DPS webové kamery Logitech 9000 Pro a systém přichycení objektivu a napájení ostření; b) Demontovaný objektiv včetně ostření z webové kamery Logitech 9000 Pro; c) Detailní pohled na CMOS čip webové kamery Logitech 9000 Pro; d) Detail IR filtru pod objektivem webové kamery Logitech 9000 Pro

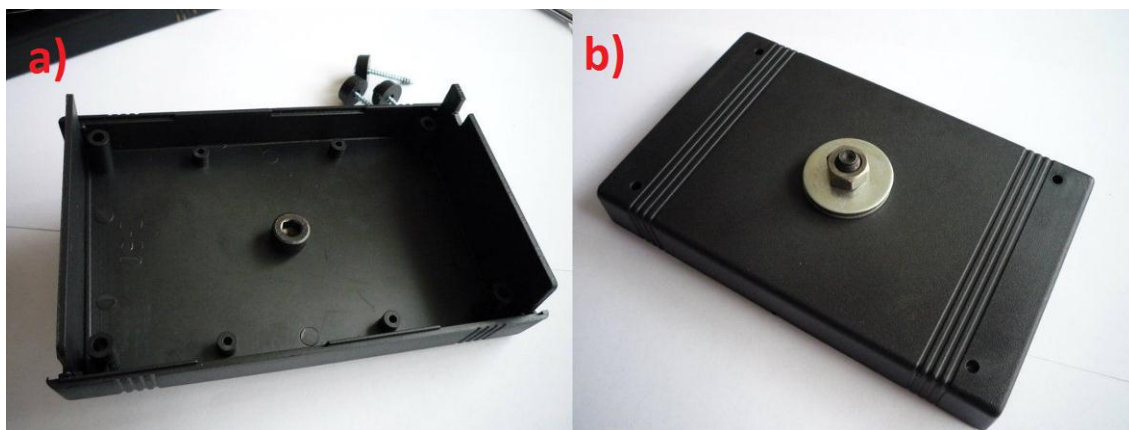
Demontovaný objektiv na obr. č. 18d) je zobrazen včetně IR filtru respektive infračervené zádrže (lesklý červený povrch), cca 1mm pod ní se nachází samotný objektiv. Tento filtr zajišťuje, schopnost kamery snímat pouze elektromagnetické záření o vlnové délce větší než je 860 nm (zkoušeno na třech druzích IR LED, 760 nm a 860 nm webová kamera ještě zaznamenala, avšak 950 nm již nikoliv). Tento infračervený filtr tudíž odřezává část elektromagnetického záření dopadajícího na CMOS čip. Jelikož bylo nutné vyměnit objektiv a tento miniaturní filtr (kruhového tvaru) fixovat na čip, rozhodl jsem se ve finální verzi tento filtr neosadit hned ze dvou důvodů. Zprv



byla komplikovaná fixace onoho filtru na nový objektiv (má větší průměr čočky) nehledě na jeho nesnadnou demontáž z původního uchycení, jeho fixace je totiž vyřešena lepidlem a zadruhé filtr lze nahradit novým i po implementaci nového objektivu ve formě šroubovacího filtru mezi objektiv a čip či násuvným filtrem přímo na vnější část objektivu. Infračervené záření bylo nutné pro nasazení webové kamery při snímání nočních scén, kde jediným osvětlovacím zdrojem je právě zdroj infračerveného světla (neinterferuje s biologickými potřebami hmyzu a navíc v něm některé druhy hmyzu fosforeskují, jsou tudíž lépe detekovatelné). Finální hardwarová úprava webové kamery tedy je schopna přijímat celé spektrum elektromagnetického záření, je však připravena pro použití rozličných a potřebných filtrů daných účelem a aplikací.

Po úpravě DPS (odstranění kabelů a objektivu s ostřením) bylo nutné zajistit vhodné usazení do nového šasi, které by mělo poskytovat větší prostor pro lepší chlazení, dostatek prostoru pro montáž uchycení závitu pro objektiv, prostor pro systém montáže samotného šasi ke konstrukci a v neposlední řadě by šasi mělo být vyrobeno z flexibilního, pružného a relativně odolného materiálu. Na stavbu šasi jsem zvolil plastovou konstrukční krabičku U-KP21 (objednanou z [www.gme.cz](http://www.gme.cz)) s rozměry (výška 46 mm, šířka 70 mm a hloubka konstrukční krabičky činila úctyhodných 120 mm. Jelikož rozměry DPS s čipem byly následující šířka 66,7 mm a výška 25,5 mm (v nejširší části DPS, její plocha totiž nemá tvar obdélníku), bylo jasné, že rozměry krabičky budou dimenzovány zcela korektně a poskytnou dostatečnou rezervu pro další montáž systému uchycení objektivu a variabilitu použití. [14]

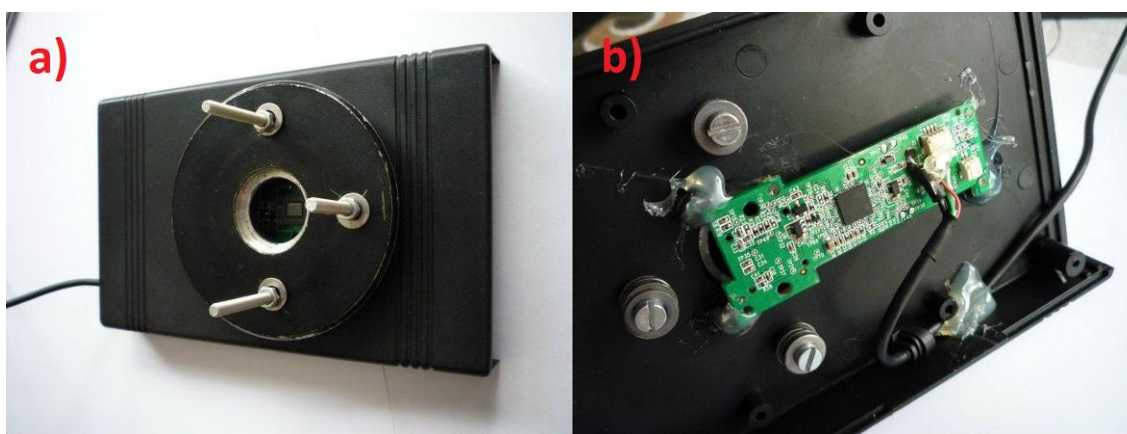
Spodní část krabičky (vnitřní část) zobrazena na obr. č. 19a) je opatřena montážním pilířem pro přichycení DPS, já však této možnosti nevyužil a DPS jsem fixoval, jak bude nastíněno dále, odlišným způsobem, nehledě na to, že pro čip webové kamery a nosné šrouby závitu objektivu, bylo nutné vyvrtat otvory, které samotný pilíř eliminovali z konstrukce (nebylo nutné ho zabrušovat do úrovně dna konstrukční krabičky). Jak je vidět na obr. č. 19b) uprostřed horní vnější části krabičky je umístěn montážní šroub s podložkou. Tento systém bude sloužit pro uchycení na rozličné konstrukce, v mém případě poslouží pro uchycení do profilované konstrukční tyče. Celé šasi drží pohromadě 4 vruty opatřené gumovými tlumícími podložkami.



Obr. č. 19 a) Náhled vnější části spodního dílu konstrukční krabičky UKP-21; b) Náhled horní části konstrukční krabičky UKP-21 se systémem uchycení (šroub s podložkou)

Do spodního dílu krabičky, vnitřní část zobrazena na obr. č. 19a) a vnější na obr. č. 19b) bylo nutné vyvrtat kruhový otvor s průměrem 40 mm pro přístup světla na CMOS čip webové kamery a 3 kruhové otvory pro nosné šrouby hliníkové redukce závitů pro objektiv o průměru 6 mm. Záměrně jsem zvolil 3 šrouby pro dokonalé srovnání rovin čipu a objektivu, DPS s čipem totiž bylo nutné umístit na dno plastové krabičky, jejíž materiál není teplotně stálý (teplem se stává pružnější) a navíc spodní část konstrukční krabičky ponese relativně velkou váhu hliníkové redukce ve srovnání se spodním víkem konstrukční krabičky, mohlo by, tudíž dojít ke kroucení částí krabičky což by mohlo mít fatální následky pro kvalitu snímaného obrazu kamery (zaostřené pouze některé části obrazu či v extrémním stavu i rozostřený celý pořízený obraz). Mezi hliníkovou redukcí závitů pro objektiv a vnější hranu krabičky jsem vložil gumové těsnění o výšce 5 mm a průměru 6 cm, které by mělo docílit absolutní izolace čipu nežádoucímu světlu z okolí (světlo musí „přicházet“ pouze objektivem). LED dioda na DPS bude vypnuta pomocí GUI ovladače dodávaného ke kameře.

Obr. č. 20b) znázorňuje samotné uchycení DPS s čipem k šasi (vnitřní hraně konstrukční krabičky), jsou použity 3 plastové distanční sloupky o výšce 4,5 mm zafixované tavným lepidlem. Tento způsob fixace není příliš standartní způsob a nepůsobí příliš esteticky, avšak vysoká pevnost spoje a možnost snadné demontáže DPS (lepidlo se fyzicky s plastovým materiálem nespojí, lze velkou silou „odloupnout“ bez destrukce materiálů) dávají možnost kdykoliv desku vyjmout či při ohřátí lepidla i dorovnat vodorovně s úrovní hrany víčka krabičky, nehledě na kvalitní fixaci a dlouhou výdrž spoje. Jediný problém byl s umístěním kabelu, který musel být krabičkou v nějakém místě protažen, zvolil sem tedy jeden z postranních bloků krabičky a vybrousil do něj čtvercový otvor pro připojovací kabel USB. Kabel jsem proti posunutí či vytržení taktéž zafixoval tavným lepidlem. Unifikovaný USB konektor šel velmi snadno připojit a zemnicí vodič bylo nutné připájet na své původní místo (obr. č. 17a)

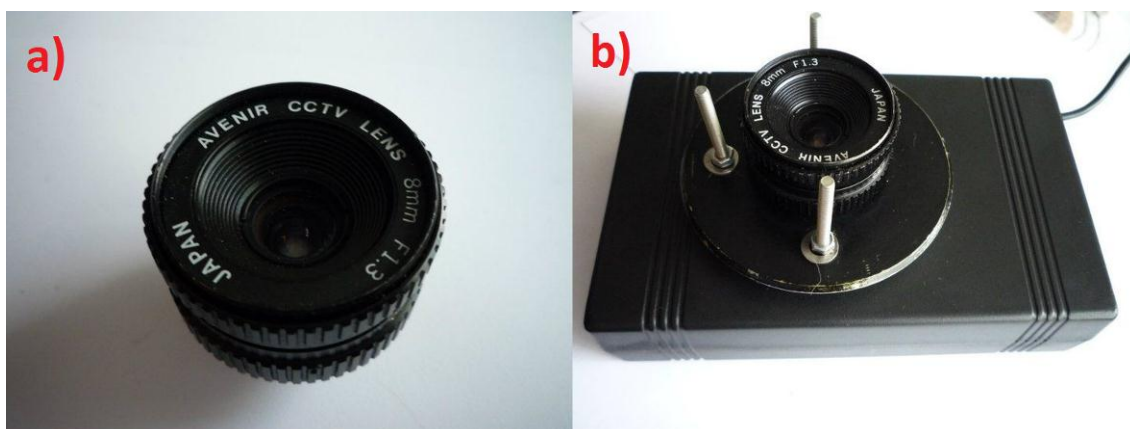


Obr. č. 20 a) Náhled vnější části spodního víka konstrukční krabičky s fixovanou závitovou redukcí pro objektiv a 3 nosnými šrouby; b) Náhled vnitřní části spodního víka konstrukční krabičky s uchycením DPS webové kamery a fixací pomocí distančních sloupků a tavného lepidla.

Samotný japonský průmyslový objektiv, na obr. č. 21a), jímž jsem nahradil původní objektiv s parametry světelnosti  $f/2.0-3.7$  (v závislosti na přiblížení), má

manuální ostření, 8 mm ohniskovou vzdálenost, má integrovanou regulovatelnou clonu a parametry světelnosti  $f/1.3$ . Pozorovací úhel činí, jak uvádí výrobce,  $45^\circ$ , což není příliš, avšak díky zúženému pozorovacímu úhlu nedochází u tohoto objektivu s malou ohniskovou vzdáleností k příliš velkému zkreslení jako u typu objektivů „rybí oko“ a byť je patrné mírné zkreslení obrazu vůči realitě, je to stále v míře únosnosti. Díky minimální objektové vzdálenosti 0,2 m a pozorovacímu úhlu objektivu bylo možné získat dostatečně široký záběr obrazu pro účely snímání vícera misek či trubiček s hmyzem (až 8 trubiček, až 4 misky) ze vzdálenosti 0,3 m. Objektiv CCTV má C systém uchycení (závitu), tento netradiční rozměr je poměrně velkou překážkou při nasazení nestandardních rozměrů filtrů (IR atd.), lze je již poměrně těžko sehnat za rozumnou cenu, proto bude asi vhodnější použít jako filtr IR světla třeba barevný osvětlený negativ filmu. [15]

Na obr. č. 21b) je již finální HW úprava webové kamery Logitech 9000 Pro, celek je poměrně masivní a dostatečně robustní pro různorodá použití oproti výrobci dodávaná forma webové kamery. Důležité je zajistit dokonalou rovnoběžnost úrovní CMOS čipu a hrany objektivu pomocí 3 nosných šroubů (přitahováním a povolováním matek).



Obr. č. 21 a) Japonský CCTV objektiv s manuálním ostřením a clonou; b) Finální podoba HW úpravy webové kamery Logitech 9000 Pro

## Použité zdroje

- [1] Logitech: Webcam Pro 9000 [online]. c2011 [cit. 2011-11-24]. Dostupné z World Wide Web:  
<<http://www.logitech.com/en-roeu/webcam-communications/webcams/devices/5867>>
- [2] Logitech: Webcam Pro 9000 [online]. [cit. 2011-11-24]. Dostupné z World Wide Web:  
<<http://i1.expansys.com/img/g/193610/logitech-webcam-pro-9000.jpg>>
- [3] CCD: Úvod do techniky CCD čipů [online]. c2011 [cit. 2011-11-24]. Dostupné z World Wide Web: <<http://ccd.mii.cz/art?id=303>>
- [4] Logitech. [program] Logitech webcam software 2.30, aktualizace z 9. prosince 2011 [cit. 2011-11-23]. Dostupné z World Wide Web: <<http://logitech.com>>. Ovladač k webové kameře. Vyžaduje Windows XP a vyšší.
- [5] Logitech support FAQ: How to get 30 Frames Per Second with your Logitech Webcam [online]. c2011 [cit. 2011-11-24]. Dostupné z World Wide Web:  
<<http://forums.logitech.com/t5/Webcams/How-to-get-30-Frames-Per-Second-with-your-Logitech-Webcam/td-p/202898>>
- [6] BŘEZINA, Březina. Fotografování: CES 2005: CMOS vs. CCD snímáče - změny na obzoru. [online]. c2009-01-14 [cit. 2011-11-24]. Dostupné z World Wide Web:  
<[http://www.fotografovani.cz/art/df\\_trendy/cmos-vs-ccd.html](http://www.fotografovani.cz/art/df_trendy/cmos-vs-ccd.html)>
- [7] Dpreview: Canon develops world's first 120 megapixel APS-H CMOS sensor [online]. c2010-08-24 [cit. 2011-11-24]. Dostupné na World Wide Web:  
<<http://www.dpreview.com/news/2010/8/24/canon120mpsensor>>
- [8] PIHAN, Roman. Fotografování: Obrazové problémy digitální fotografie I. - Senzor: Schématické znázornění jedné buňky CMOS senzoru [online]. c2011-03-2009 [cit. 2011-11-24]. Dostupné na World Wide Web:  
<[http://www.fotografovani.cz/images3/rom\\_trouble1\\_07.jpg](http://www.fotografovani.cz/images3/rom_trouble1_07.jpg)>
- [9] ŠURKALA, Milan. Digimanie: Jak si vybrat kompaktní: snímací čipy [online]. c2011-05-27 [cit. 2011-11-24]. Dostupné na World Wide Web:  
<[http://www.digimanie.cz/art\\_doc-7051479ADC3A5D5CC125772A007AE74F.html](http://www.digimanie.cz/art_doc-7051479ADC3A5D5CC125772A007AE74F.html)>
- [10] FORRET, Petter. [program] Online Video Bitrate calculator, [cit. 2011-11-24]. Dostupné na World Wide Web: <[http://web.forret.com/tools/video\\_fps.asp](http://web.forret.com/tools/video_fps.asp)>. Online kalkulačka. Vyžaduje internetový prohlížeč.
- [11] Dagineff: Co je interpolace [online]. c2001-10-26 [cit. 2011-11-24]. Dostupné na World Wide Web: <<http://dagineff.cz/cojeto/ruzne/interpolace.html>>
- [12] Paladix: Co se děje, když se obrázky zmenšují a zvětšují [online]. c2001-12-10 [cit. 2011-11-24]. Dostupné na World Wide Web:  
<<http://www.paladix.cz/clanky/co-se-deje-kdyz-se-obrazky-zmensuji-a-zvetsuji.html>>

- [13] Logitech: FOV - EFL of 9000 Pro [online]. c2011 [cit. 2011-11-24]. Dostupné na World Wide Web: <<http://forums.logitech.com/t5/Webcams/FOV-EFL-of-9000-Pro/td-p/293076>>
- [14] GM electronic: U-KP21 [online]. c2011 [cit. 2011-11-24]. Dostupné na World Wide Web: <<http://www.gme.cz/plastove-krabicky-univerzalni/u-kp21-p622-422/>>
- [15] Datasheet: Avenir CCTV (MTCH0813) [online]. [cit. 2011-11-24]. Dostupné na World Wide Web: <<http://www.sourcesecurity.com/docs/fullspec/MTCH0813-1.pdf>>

## 4 SW ŘEŠENÍ

### 4.1 Volba vývojového prostředí

Více než samotné programovací prostředí, bylo důležité vybrat vhodný programovací jazyk. Díky své dlouholeté zkušenosti s programovacími jazyky na bázi BASIC jsem se rozhodl pro moderní verzi programovacího prostředí Visual Studio .NET verze 2008. Toto prostředí poskytuje dostatečnou základnu pro různé programovací jazyky (Visual C++, Visual C#, Visual Basic .NET a další v závislosti na verzi a účelu programovacího prostředí). Výhodou použití tohoto programovacího prostředí jsou:

- Přímá tvorba grafického rozhraní programu a zdrojového kódu zároveň
- Dokonalý online systém nápovědy (MSDN podpora)
- Možnost integrace již hotových modulů, knihoven a komponent
- Možnosti kompilace pro různé typy systémů (X86, X64)
- Funkce a schopnosti platformy .NET Framework

Mým zvoleným programovacím jazykem se stal Visual Basic .NET, jelikož s tímto jazykem mám dlouholeté zkušenosti. Tento jazyk poskytuje dle [1] zejména začátečníkům snadný a rychlý postup ve vývoji software, jeho příkazy se totiž skládají zpravidla ze zkratk anglických slov popisujících příkazy (např. příkaz DIM pro deklaraci proměnné je zkratkou anglických slov „Dimension In Memory“, které v překladu znamenají prostor v paměti – který proměnná, jejíž název je následuje za příkazem, obsadí). Visual Basic .NET je postaven na platformě .NET Framework, kterou ke svému chodu vyžaduje nainstalovanou v operačním systému Windows, ale také na cílovém PC kde bude program spouštěn. Jelikož je platforma .NET Framework dostupná v současné době pouze pro OS Windows (stala se nedílnou součástí operačního systému Windows buď přímo v instalaci či jako dodatek ve formě aktualizací přes Windows Update), není možné hotové kompilované programy vytvořené ve Visual Basic .NET spouštět na jiných operačních systémech (OS Linux, Mac OSX atd.). Tento fakt ovšem nebyl důležitý, jelikož vyvíjený software měl běžet na operačním systému Windows (pro snadné zpracování dat obsluhou). Jak uvádí zdroj [1], na systému konverze těchto programů i pro ostatní operační systémy, kde defaultně neobsahuje platformu .NET Framework, se pracuje a v budoucnu bude takto vyvíjené programy možné použít bez omezení.

Visual Studio 2008 v sobě integruje také systém pro generaci instalačního balíčku výsledného vyvíjeného programu, ať už pomocí off-line distribuce („\*.exe“ či „\*.msi“ soubor), kterou lze přenášet na vhodně velkém médiu, tak také online distribuci (instalace dostupná z FTP, webových stránek apod.), čímž odpadly dodatečné případné náklady na pořízení programu pro tvorbu instalátoru.

Prvotní záměr vyvíjet program jako multiprocesorový (zpracování by prováděl víceprocesorový PC) byl z ne příliš komfortního řešení ovládání a řízení

multithreadingových (více-vláknových) procesů pomocí Visual Basic .NET verze 2008 (postaveno na .NET Framework 3.5) zrušen, v nedávno vyšle verzi Visual Basic .NET 2010 (postaveno na platformě .NET Framework 4) je již více-vláknové zpracování řešeno excelentně avšak datum vydání této verze přišlo právě v době již rozpracovaného programu a nebylo možné snadným způsobem konvertovat k této nové verzi jazyka.

Vývojové prostředí obsahující mimo jiné také Visual Basic .NET je zcela zdarma a je možné ho zcela zdarma a legálně stáhnout z internetového zdroje výrobce (<http://www.microsoft.com/cze/msdn/vstudio/2010/default.aspx>) ve své verzi Express Edition. Já jsem však použil profesionální edici šířenou pod školní licenci MSDN (Visual Studio Professional 2008) obsahující Visual Basic .NET 2008, tudíž mé náklady na pořízení tohoto vývojového software byly nulové.

## 4.2 Výběr názvu vyvíjeného software

Díky inspiraci v názvech konkurenčních programů (TriKinetics, CTRAX, PySolo) zabývajících se obdobnou problematikou, jsem pro svůj vyvíjený software zvolil taktéž název ve formě zkratky, jejíž definice popisuje cíl a práci software. Název mého software je „DROSANA“, což je zkratkou anglických slov „Drosophila analyzátor“.

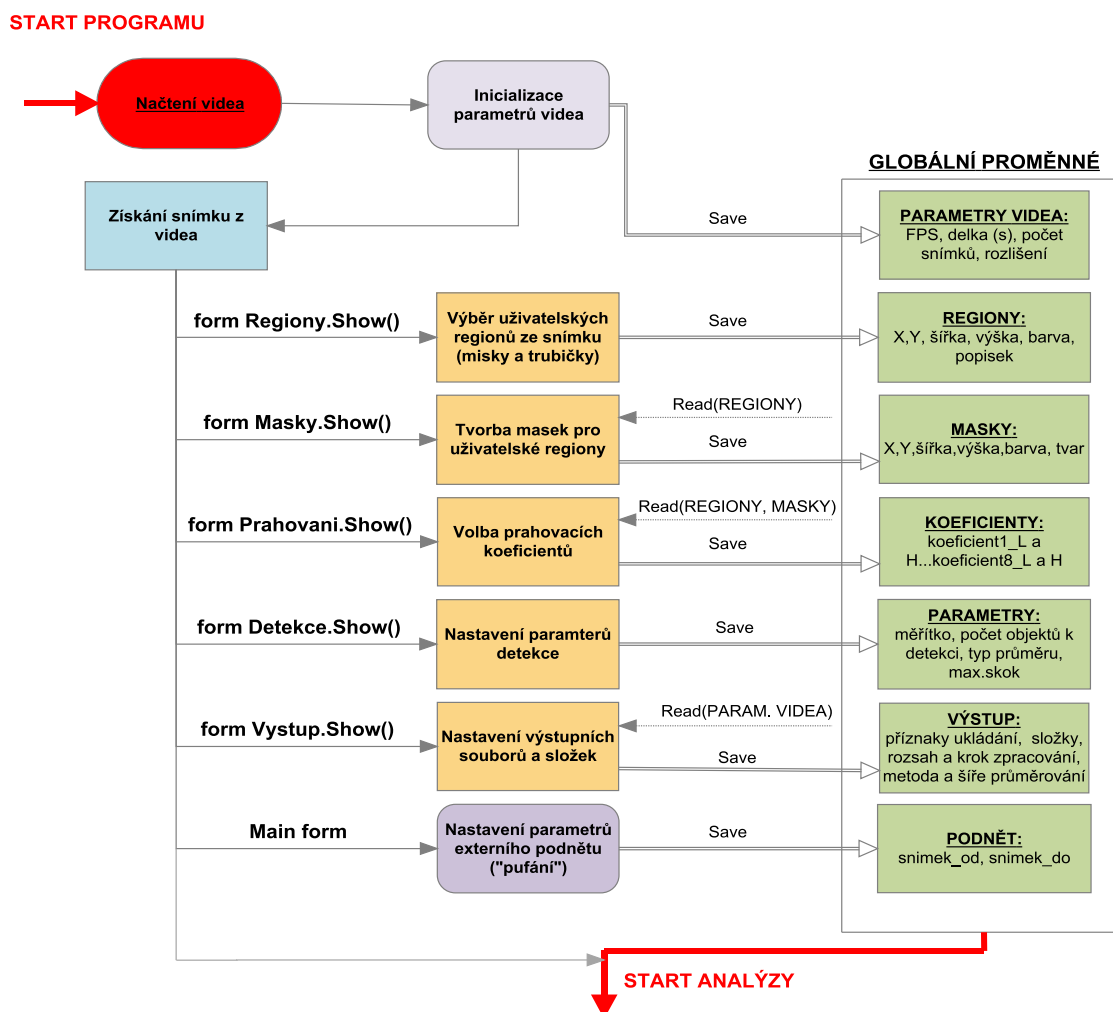
## 4.3 Návrh grafického rozhraní software

Na obr. č. 1 je znázorněn vývojový diagram a organizační struktura grafického rozhraní první beta verze software DROSANA.

Rozhraní je realizováno pomocí několika formulářů, které jsou řízeny uživatelským výběrem. Hlavní formulář, který je otevřen defaultně ihned po spuštění programu jako první, slouží pro **načtení videa**. Dalším krokem, který proběhne na pozadí automaticky po načtení videa je **inicializace informací o načteném videu** (rozlišení videa, FPS, délka (snímků), délka (s)). Tyto informace jsou pak dále využívány pro výpočty výsledných dat a omezující limity grafických nastavení.

Komponenta „MCIWndX Control“ v1.0 (vyrobena v roce 1996 firmou Microsoft, autor Simon Bernstein), která umožňuje načtení a práci s videem (otevřít je možné video ve formátech AVI a WMV), je použita záměrně. Schopnosti této komponenty jako její snadná implementace (obsahuje grafické rozhraní pro správu otevřeného videa), její schopnost přímo ze zdrojového kódu ovládat a kontrolovat činnost přehrávaného videa (rychlost přehrávání, ovládání hlasitosti, zoom) a hlavní schopnost programového řízení možnosti pohybu po jednotlivých snímcích (vpřed i vzad) včetně **kopírování zobrazeného snímku videa** do jiné komponenty ji předurčily k mému použití, bylo totiž potřeba získat každý jednotlivý snímek videa, který bylo nutné dále upravit a pomocí obrazových algoritmů analyzovat.

Získané snímky z videa poskytly dostatečné vstupní materiály pro jednotlivá přednastavení obrazové analýzy. Tyto přednastavení (defaultně nastaveny na optimální či nulové hodnoty) se ukládají do **globálního úložiště proměnných**, které je dostupné pro čtení a zápis napříč celému projektu se zdrojovým kódem.



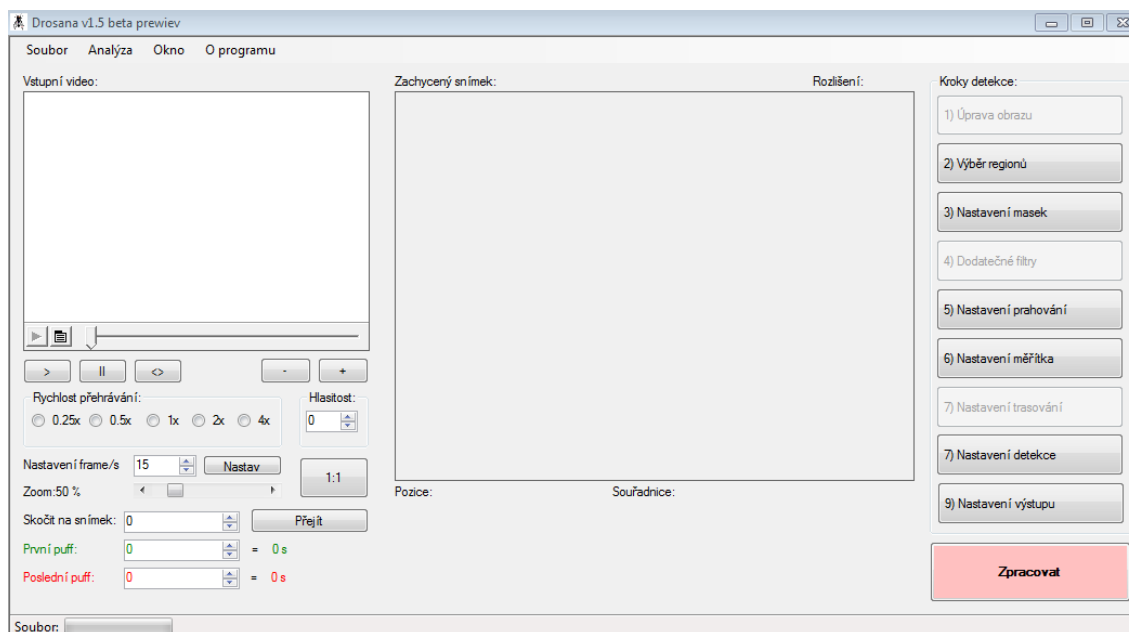
Obr. č. 1 Vývojový diagram grafického rozhraní vyvíjeného software DROSANA

Přednastavení se provádí pomocí série několika oddělených a na sobě závislých **formulářů** (na obr. č. 1 označeny oranžovou barvou), které je možné zobrazovat („volat“) z hlavního formuláře popořadě pomocí tlačítek či pomocí kontextového menu v horní části hlavního formuláře jak je vidět z následujícího obr. č. 2.

Řízení formulářů bude řešeno pomocí třídy Form (ve jmeném prostoru System.Windows.Forms). Formulář je vlastně kontejner, který obsahuje vhodně rozmístěné ovládací prvky programu tak aby umožňovali uživateli snadnou manipulaci a nastavení programu. Program bude obsahovat hlavní MDI formulář (typ formuláře, který se v projektu nachází jediný a vlastní další formuláře). [7]

Samotné rozvržení hlavního formuláře programu DROSANA se skládá z několika částí. Levá část formuláře obsahuje prvky zobrazující a ovládající načtené video, střední část zobrazuje aktuální snímek videa včetně jeho parametrů a levá část formuláře obsahuje výše zmiňovanou sérii tlačítek, které volají zobrazování jednotlivých formulářů. Byť jsou formuláře očíslovány, není je nutné zobrazovat v hierarchickém pořadí daném jejich čísly.





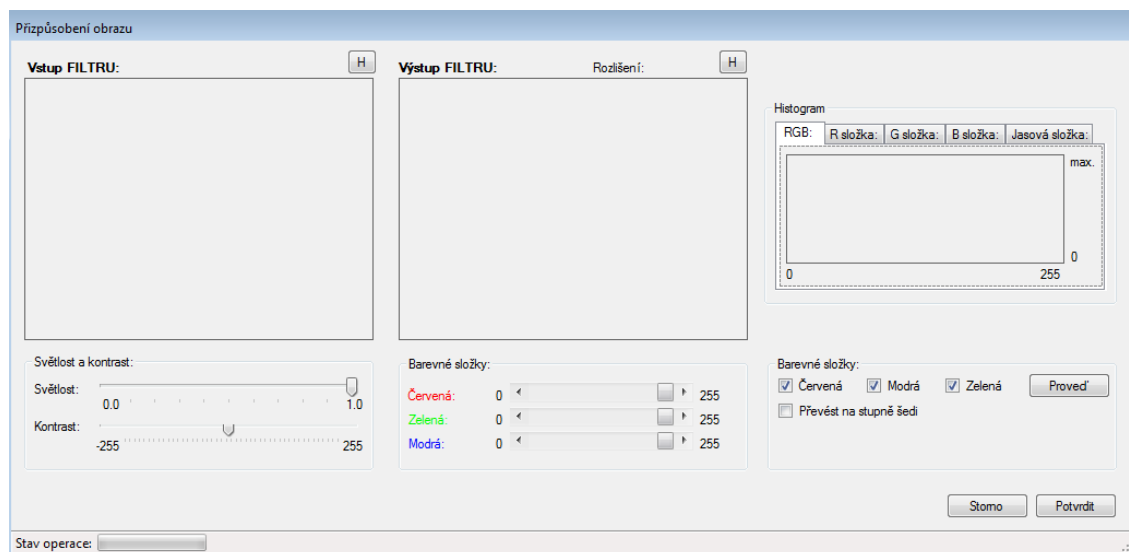
Obr. č. 2 Hlavní formulář programu DROSANA

## 4.4 Popis formulářů programu DROSANA

### 4.4.1 Úprava obrazu

První dostupný formulář s pořadovým číslem „1“, je formulář s názvem „Úprava obrazu“. V původní vznikající beta verzi 1.0 programu DROSANA byl tento formulář dostupný, avšak díky faktu, že video je analyzováno off-line (tzn. ne přímo při natáčení), je možné přistoupit k sofistikovanějším úpravám samotného videa v programech, které jsou nabízeny zdarma či komerčně (Sony Vegas, Adobe After Effects, Windows Movie Maker, Virtual Dub a jiné.), což byl důvod, proč jsem se rozhodl od verze 1.1 tento formulář defaultně zakázat (byť je stále implementován v programu).

Samotný formulář zobrazený na obr. č. 3, u kterého jsou zakázány ovládací prvky (minimalizace, maximalizace, zavření), se skládá ze dvou náhledových boxů pro obrázek a ovládacích prvků, slouží k úpravě snímků (bitmap) získaných z videa, umožňuje pomocí táhel nastavit světlost obrazu, kontrast a intenzitu jednotlivých barevných složek RGB (včetně eliminace některé ze složek RGB). Statistika a histogram se zobrazuje ihned po načtení obrazu a dále pak při jakékoliv uživatelské změně. Levý box pro obrázek obsahuje vstupní obrázek videa (tzn. snímek zkopírovaný z videa), levý box pak slouží pro zobrazení levého obrazu s aplikovanými úpravami obrazu, náhled formuláře je vidět na obr. č. 3)



Obr. č. 3 Formulář „Úprava obrazu“ programu DROSANA

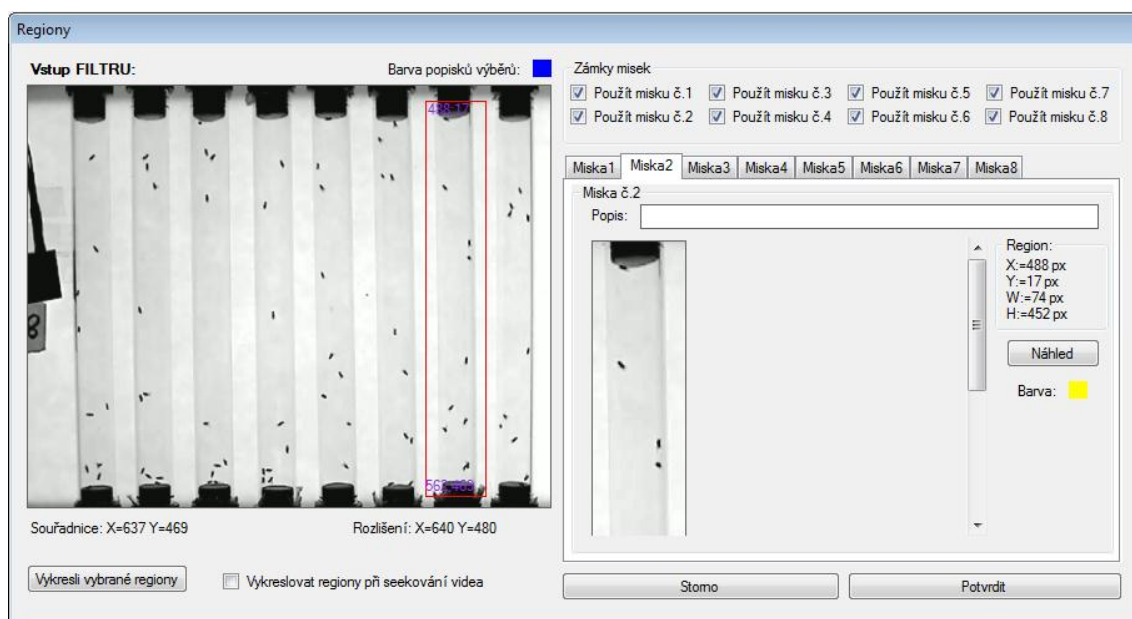
#### 4.4.2 Výběr regionů

Další formulář nazvaný „Výběr regionů“, slouží k jediné věci, a to k uživatelskému výběru jednotlivých užitečných částí snímku (obrazu) z videa, které tvoří unikátní či jedinečný děj. Celý formulář je rozdělen na levou část, která se skládá z velkého boxu pro obrázek (zde se po načtení nachází celý snímek z videa) a dále pak pravé části, kde se nachází jednotlivé prvky pro zobrazení a úpravu (pojmenování vybraného regionu, nastavení barvy vykreslování okraje regionu) uživatelsky vybraných regionů. Region je část (výřez) původního snímku z videa, je proto možné vybrat jak kruhové misky, tak i podlouhlé obdélníkové trubičky.

Na obr. č. 4 jde např. o výběr jednotlivých trubiček ze snímku videa, na kterém zachyceno celkem 8 trubiček najednou. Maximální počet regionů, které je program DROSANA schopen analyzovat je 8, avšak stejné děje je možné sdružit a označit jako jeden region, tzn., že je možné označit 2 a více trubiček a ve výsledné analýze poté sledovat separovaná data jednotlivých much, byť jde o jediný region, tudíž je možné teoreticky použít neomezené množství regionů (v závislosti na kvalitě videa, velikosti objektů).

**Postup:** Uživatel klikne levým tlačítkem myši do prostoru, kde se nachází box s obrázkem (snímek z videa), zde za pomoci tažení a stálého držení myši označí region, který chce z obrázku separovat (označením části obrázku dojde k automatickému zkopírování této části do paměti). Poté povolí libovolný zámek misky v horní pravé části formuláře, přesune se na příslušné kartě na požadovanou „virtuální“ misku a levým tlačítkem myši klikne do prostoru náhledu (do prostoru prázdného obdélníku), tím se výsledná část obrazu (region) zkopíruje do příslušné „virtuální“ misky a vytvoří tak region. Uživatel dále může region opatřit specifickým popisem v textovém poli „Popis“ (defaultně i při prázdném poli „Popis“ vystupuje „virtuální“ miska pod názvem ve formát „MiskaX“, kde X označuje číslo misky). Další možnosti

nastavení regionu je barva, po kliknutí na barevný čtverec znázorňující aktuální barvu regionu je možné vybrat jinou než defaultní (žlutou) barvu okraje regionu. Tato barva nemá vliv na analýzu, slouží pouze k vizuální separaci odlišných regionů (po kliknutí na tlačítko „Vykresli vybrané regiony“ se v hlavním boxu se snímkem videa vykreslí barevné hrany vybraných regionů, stejného efektu je možné docílit také v hlavním okně programu DROSANA při krokování videa pomocí zaškrťovací položky „Vykresli vybrané regiony při seekování videa“). Celý region je možné zobrazit pomocí tlačítka „Náhled“, zde je ho možné poté uložit jako bitmapu na paměťové zařízení (HDD apod.). Uložení výběru regionů se potvrdí (uloží do paměti) tlačítkem „Potvrdit“.



Obr. č. 4 Formulář „Výběr regionů“ programu DROSANA

**Technické informace:** Uživatel výběry jednotlivých požadovaných regionů vlastně předpřipraví pouze ty části obrázku, které jsou užitečné, tím se mnohonásobně urychlí celý proces analýzy (regiony mají menší rozlišení, tudíž není potřeba procházet bitmapy s velkým počtem pixelů). Každý uživatelsky vybraným region je uložena do globálních proměnných ve formě použití či nepoužití regionu (hodnota 0 či 1), informace o počátečních souřadnicích regionu (X, Y) v px (typ Integer), dále šířce výběru W (width) v px (typ Integer), výšce regionu H (height) v px (typ Integer), barvě (typ Color) a popisku regionu (typ String).

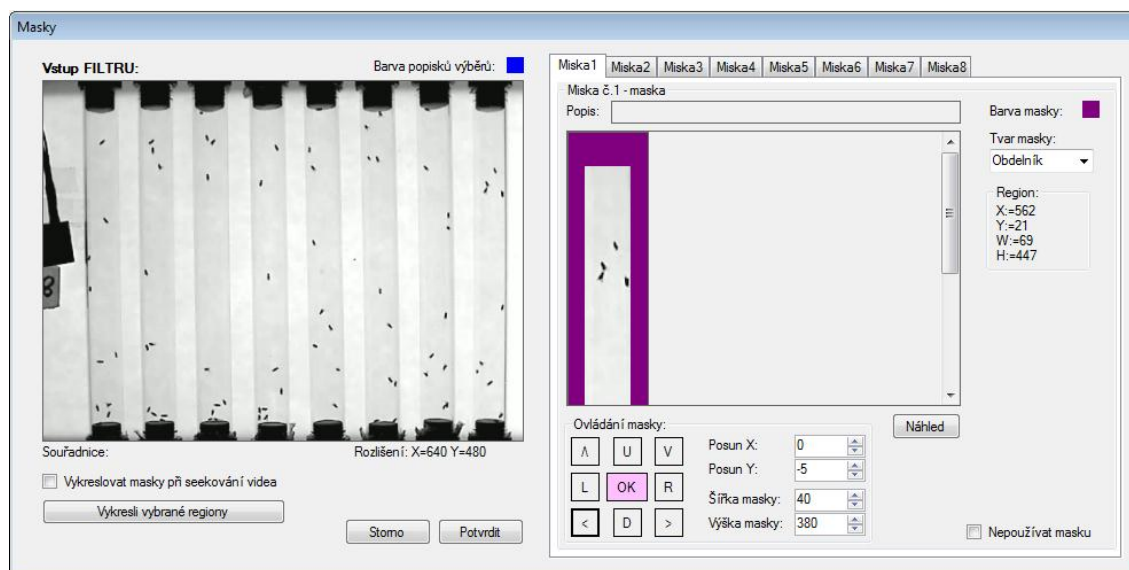
#### 4.4.3 Nastavení masek

Formulář s názvem „Nastavení masek“ je podmíněn nastavenými údaji o uživatelsky vybraných regionech, tzn., že ho nelze zobrazit bez předchozího vybrání a nastavení uživatelských regionů (viz. 4.4.2 Výběr regionů).

Formulář se skládá z velkého náhledového okna boxu s obrázkem, jak je znázorněno na obr. č. 5, který obsahuje původní snímek z videa, a ovládacích prvků, každé jednotlivé „virtuální“ misky (pravá část formuláře). V misce se nachází uživatelský region, čili výřez z obrázku vlevo, na který je aplikována maska, která zakryje nedostatky v obraze (odlesky, hrany, konstrukci apod.).

**Postup:** Po načtení formuláře jsou automaticky povoleny pouze ty „virtuální“ misky, jež nejsou prázdné, ostatní jsou nastaveny programově na „disabled“. Uživatel se přepne na „virtuální“ miskou, se kterou potřebuje pracovat a zvolí si pomocí rozbalovacího menu „Tvar misky“ jaký tvar misky potřebuje. Na výběr je „Kruh“ pro misky a „Obdélník“ pro trubičky. Po tomto výběru, respektive už od spuštění formuláře je vygenerována defaultní maska, která je nastavena na rozměry, jejíž šířka odpovídají 15 % výšky regionu a 15 % šířky regionu. Rozměry masky a samotný posun masky je možné upravit pomocí ovladače „Ovládání masky“. Ovládání je umožněno buď manuálně, tzn. zadáním přesné hodnoty do polí „Posun X“ = posun masky v horizontálním směru, „Posun Y“ = posun masky ve vertikálním směru, „Šířka masky“ = změna šířky masky v ose x, „Výška masky“ = změna šířky masky v ose y, je vždy ale potřebné vykreslení masky obnovit pomocí tlačítka „OK“ ve středu křížového ovladače, který slouží k jemnému dokorigování masky jen pomocí myši. Tlačítka „<“ a „>“ ovlivňují šířku masky v ose x, tlačítka „^“ a „v“ ovlivňují šířku masky v ose y, tlačítka „U“ a „D“ ovlivňují posun masky ve vertikálním směru („U“ – up-nahoru, „D“ – down-dolů), tlačítka „L“ a „R“ ovlivňují posun masky v horizontálním směru („L“ – left-doleva, „R“ – right-vpravo). Barva masky je defaultně nastavena na fialovou barvu, lze ji změnit kliknutím na barevný čtverec u popisu „Barva masky“. **POZOR! Barva masky ovlivňuje analýzu snímku, nikdy by neměla být nastavena na černou barvu či takovou, jež se vyskytuje v obraze! Masku je potřeba nastavit velmi přesně tak, aby plocha masky přesně překrývala nežádoucí objekty, hrany a jiné části v obraze regionu.** Výsledný region s aplikovanou uživatelskou maskou lze opět jako v případě formuláře pro tvorbu regionů zobrazit pomocí tlačítka „Náhled“. Nastavení masek lze uložit do paměti pomocí tlačítka „Potvrdit“. Formulář nastavení masek je zobrazen na obr. č. 5.

**Technické informace:** Nastavením a aplikací masky uživatel způsobí přesné překrytí drobných nežádoucích objektů v obraze regionu (např. hran, nežádoucích cizích objektů apod.). Díky tomu je již možné spolehlivá detekce objektů v obraze. Maska slouží také jako korekční plocha při prahování, region (užitečný výřez ze snímku videa) je totiž vyprahován na bitovou bitmapu, která obsahuje pouze černou a bílou barvu (odlišení pozadí a popředí), po nanesení masky (barevné) je možné detekovat pouze užitečné objekty v obraze (užitečné objekty zůstanou černé), což by v případě nepoužití masky bylo obtížné, poněvadž by v obraze vystupovaly jako černé plochy i hrany či okolí regionu, které by mohly komplikovat detekci objektů (mohli by vystupovat jako objekty). Nastavení masek je ukládáno do paměti pomocí parametrů: barva masky (typ Color), tvar masky (typ Byte), posun masky v ose X (px) a ose Y (px) oba posuny jsou typu Integer, šířka masky (px) typ Integer, výška masky (px) typ Integer.

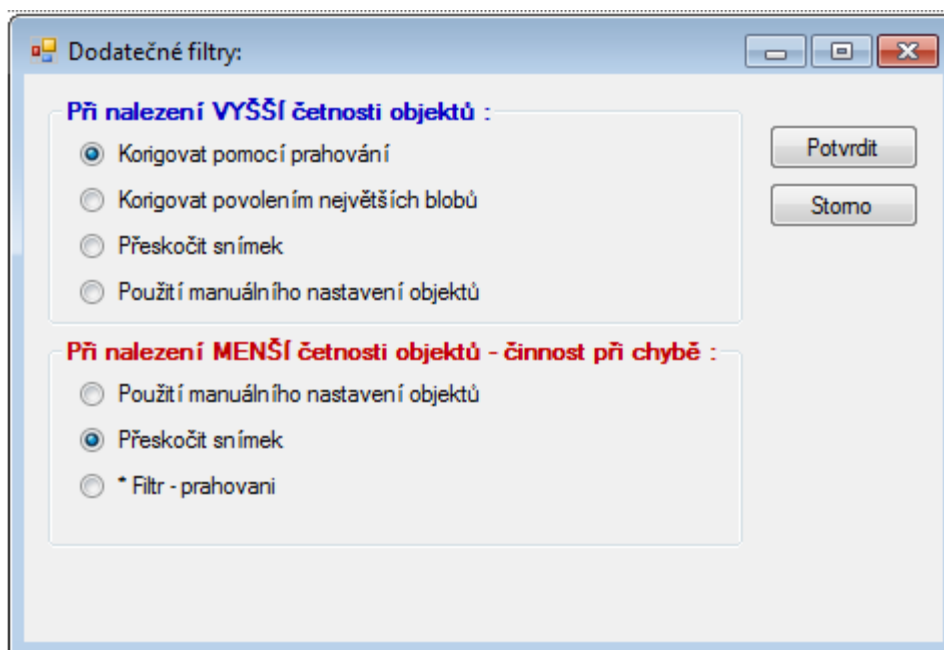


Obr. č. 5 Formulář „Nastavení masek“ programu DROSANA

#### 4.4.4 Dodatečné filtry

Formulář s názvem „Dodatečné filtry“ slouží k nastavení příznaků pro používání různých filtrů, které se mají aplikovat při nekorektní detekci (není nalezen korektní nastavený počet objektů v obraze).

Skládá se z nastavení pro vyšší nalezený počet snímků (tzn. některý z objektů je v obraze navíc, či vznikl při úpravě obrazu konkrétního regionu) a nastavení pro nižší než nastavený počet objektů (tzn., že některý z objektů v obraze se skryl za jiný viditelný objekt, či objekt zmizel při úpravě obrazu). Filtry „Korigovat pomocí prahování“ a „Filtr-prahování“ jsou používány zejména při eliminaci různých nežádoucích stavů, jako jsou rozpady much vlivem prahování, šumu v obraze či přesvětlení části obrazu, filtry „Použit manuálního nastavení objektů“ pak umožňuje vypnutí detekčních částí software DROSANA a manuálního určení objektů pomocí dodatečného formuláře který se zobrazí u každého snímku, kde není nalezen nastavený počet objektů detekce. Přeskakovat „špatně detekované“ snímky lze pomocí volby „Přeskočit snímek“, snímek bude přeskočen a ve výsledných datech bude záznam o přeskočení snímku. Defaultně je tento formulář od verze 1.2 programu DROSANA nastaven programově na „disabled“, jelikož hlavní detekční algoritmus byl vylepšen natolik, že již těchto filtrů není potřeba (krom manuální identifikace objektů kterou řídí filtr „Použit manuálního nastavení objektů“, který je velmi důležitý a užitečný, ale od verze 1.2 již není dostupný, od verze 1.6, již tento filtr bude opět možné nastavit).



Obr. č. 6 Formulář „Dodatečné filtry“ programu DROSANA

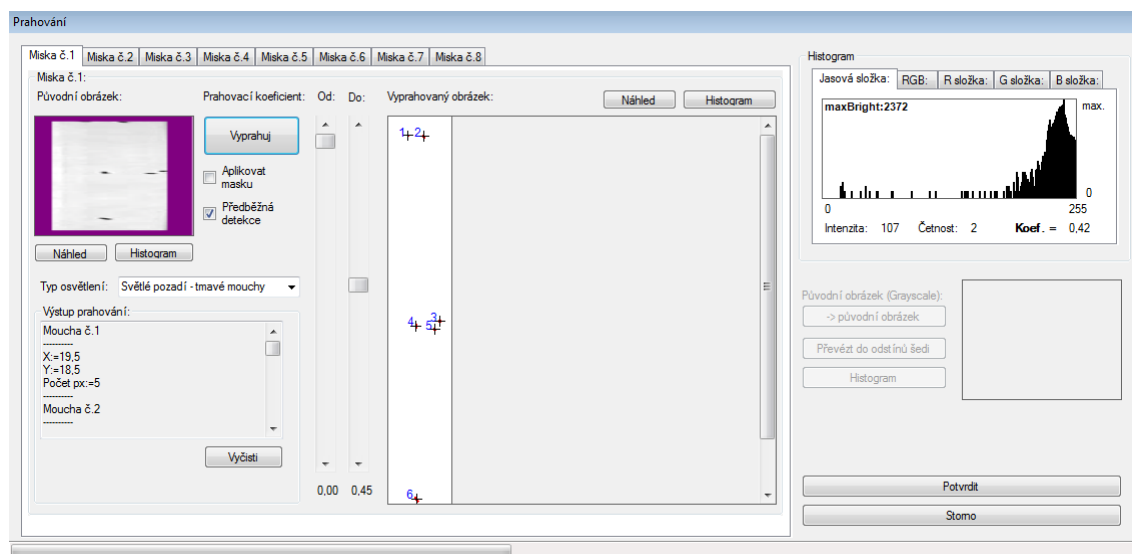
(\* Filtr „Prahování“ či „Korigovat pomocí prahování“ pracuje na následujícím principu: k prahovacím koeficientům jsou dle výsledné detekce buď přičítány či odčítány koeficienty 0,05, čímž se velmi zjemní chod prahování (defaultně jde o 0.5) a je tak možné s velkou pravděpodobností možné získat korektně detekovaný snímek, bohužel někdy na úkor „rozpadu“ či „slití“ jiných objektů).

#### 4.4.5 Nastavení prahování

Pravděpodobně jeden z nejsložitějších a nejdůležitějších formulářů software DROSANA se nazývá „Nastavení prahování“. Zobrazení formuláře je podmíněno přednastavením regionů a masek, bez těchto nastavení není možné tento formulář zobrazit. Formulář sám o sobě slouží k nastavení dvojice prahovacích koeficientů (dolní a horní hranice) vždy pro jednu „virtuální“ misku, respektive region (výřez z původního snímku videa). Region je pak dle těchto koeficientů (koeficienty určují hodnoty jasu pixelů, které mají, budou „očerněny“, vše ostatní bude „obíleno“) prahován na černobílý obraz tak, že popředí splňující nastavené koeficienty (objekty) jsou černé a pozadí (bílé).

**Postup:** Po načtení formuláře je nutné, aby uživatel klikl na tlačítko „Histogram“ nacházející se pod náhledem regionu s aplikovanou maskou. Výsledkem je vykreslení histogramu obrázku (vpravo nahoře formuláře), který je potřeba převést na 1 bitovou bitmapu. Samotné nastavení prahovacího koeficientu je nutné vyčíst právě z histogramu. V závislosti na konkrétním obraze (regionu s aplikovanou maskou) je možné předpokládat, že velkou část obrazu tvoří pixely podobné barvy (světlá barva), tudíž největší četnosti hodnoty jasu v obraze bude dosahovat právě pozadí. Na histogramu je pozadí tedy znázorněno největším tzv. peakem, kolem nějž se soustředují s poměrně velkou četností (již však menší) pixely podobného jasu (světlé). Aby bylo

možné oddělit pozadí (osvětlená miska či trubička) a popředí je nutné nastavit „Prahovací koeficient do“ tak, aby se zelená čára v podobě ukazatele (zobrazovaná při změně výše uvedeného posuvníku) držela právě před či za největšími „peaky“ (čili pozadím), v případě na obr. č. 7, je nutné ukazatel nastavit s mírnou rezervou před shluk největších „peaků“, což odpovídá cca 0,55 (hodnota jasu 140 z maximálních 255). „Prahovací koeficient od“ je možné ponechat v základním nastavení 0,00, vytvoří se tak rozsah jasů (rozsah=Prahovací koeficient od – Prahovací koeficient do) jež bude popisovat strukturu pixelů popředí (objektů). V závislosti na základní úrovni jasu videa lze defaultně používat tyto nastavení: Prahovací koeficient od = 0, Prahovací koeficient do = 0,5. Korekci prahovacích koeficientů je nutné ověřit pomocí histogramu (návod výše). Jsou-li nastavení koeficientů hotová, je možné přistoupit k ručnímu a orientačnímu ověření detekce objektů pomocí tlačítka „Vyprahuj“. To způsobí, že za pomoci nastavených prahovacích koeficientů ke konkrétní „virtuální misce“ s regionem a aplikovanou maskou, je obraz ve „virtuální misce“ převeden na černobílý (1 bitový), kde popředí tvoří objekty (černé) a pozadí (osvětlená miska či trubička). V náhledovém okně (uprostřed formuláře) je možné sledovat jak takový vyprahovaný (1 bitový) obrázek vypadá. Je nutné zkontrolovat, zda v obraze nejsou nežádoucí pixely („roztřepené okraje“, šumové pixely kolem objektů apod.), pokud ano, je nutná korekce prahovacích koeficientů (snížení, zvýšení) s opakováním procesu prahování. Je-li vše v pořádku, je možné detekci opakovat se zapnutou volbou „Aplikuj masku“ (na výsledný vyprahovaný obrázek použije barevnou masku) či volbou „Předběžná detekce“ (výsledný vyprahovaný obrázek je podroben předběžné obrazové analýze nalezením objektů včetně jejich středů, celý výstup je pak doprovázen textovými daty v poli „Výstup prahování“, které obsahuje identitu objektu, souřadnice středu a počet pixelů které jej celkově tvoří, v závěru je pak celková suma nalezených objektů). Textový výpis výstupu prahování je možné vymazat kdykoliv tlačítkem „Vyčisti“. Uložení hodnot nastavených prahovacích koeficientů je možné pomocí tlačítka „Potvrdit“.



Obr. č. 7 Formulář "Nastavení prahování" programu DROSANA

**Technické informace:** Dvojice prahovacích koeficientů vždy pro konkrétní „virtuální miskou“ je nutná pro převod jednotlivého regionu s aplikovanou maskou na 1 bitovou černobílou bitmapu, která pak finálně slouží jako zdroj pro detektor blobů (užitečných objektů). Nastavení koeficientů je tak velmi důležité pro následnou analýzu a detekci. Celkově program umožňuje uložit 16 koeficientů (dvojice koeficientů = dolní a horní hranice pro 8 „virtuálních“ misek). Programově jsou vedeny jako typ proměnné Single (jde o desetinná čísla) a defaultně nastaveny na hodnoty: dolní hranice 0,00 a horní hranice 0,45.

#### 4.4.6 Nastavení měřítka

Tento nezávislý formulář s názvem „Nastavení měřítka“ zobrazený na obr. č. 8 slouží pro nastavení programového parametru, který uchovává informaci počtu pixelů připadajících na 1 mm skutečné délky. Tento parametr je důležitý při přepočtu naměřených hodnot analyzovaných objektů na snadno představitelné a porovnatelné hodnoty (odvozené jednotky délky SI)



Obr. č. 8 Formulář „Nastavení měřítka“ programu DROSANA

**Postup:** Uživatel po načtení formuláře klikne v obrazovém boxu, který zobrazuje původní snímek z videa, levým tlačítkem myši na počáteční hranu předem stanoveného objektu, o němž zná rozměry a za stálého držení levého tlačítka myši táhne označovací linku až po koncovou hranu tohoto objektu. Program počítá vzdálenost, respektive počet pixelů, které tuto linku tvoří a zobrazuje tento počet v popisku „Délka“. Dalším krokem uživatele je vyplnění onoho známého rozměru objektu

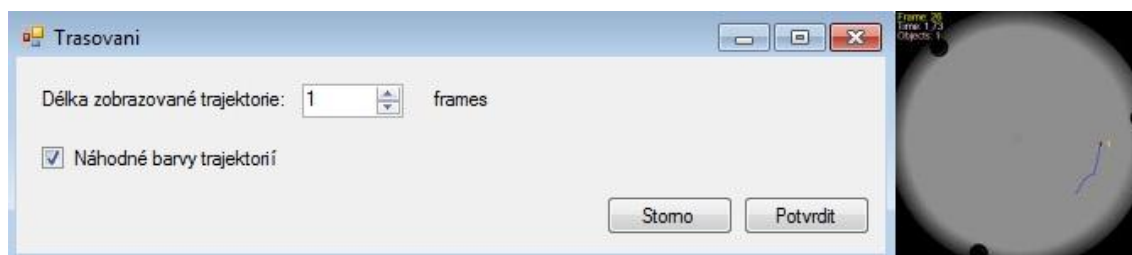


do textového pole. Pro uložení a vypočtení poměru pixelů na 1 mm je potřeba stisknout tlačítko „Potvrdit“, čímž se zapíše tento poměr do proměnné v paměti.

**Technické informace:** Počet pixelů připadajících na 1 mm je vypočten na 5 desetinných míst. Výsledný poměr je proměnná typu Double a je pak dále používán při převodu dráhy, rychlost a zrychlení na odvozené jednotky SI.

#### 4.4.7 Nastavení trasování

Formulář s názvem „Nastavení trasování“ je jedním z nejjednodušších formulářů v programu DROSANA. Slouží k nastavení velikosti trasovacího pole (parametr  $n$ ). Toto pole uchovává středy jednotlivých much v průběhu  $n$  analyzovaných snímků videa. Obsah tohoto pole je pak používán při vykreslování jednotlivých výsledků do analyzovaných snímků, čímž se docílí zajímavého efektu zobrazení jednotlivých trajektorií detekovaných objektů.



**Obr. č. 9** Vlevo: Formulář "Nastavení trasování" programu DROSANA; Vpravo: Výstupní obrázek analýzy se zapnutým trasováním.

**Postup:** Po načtení formuláře vyplní uživatel v poli „Délka zobrazované trajektorie“ hodnotu  $n$ , která reprezentuje počet analyzovaných (souřadnic středů) z  $n$  snímků. Tyto hodnoty jsou pak v každém snímku vykreslovány pro každý nalezený objekt, vytváří se tudíž trajektorie pohybu, u které je možné nastavit délku (v jednotkách snímků). Díky zaškrtnutí položky „Náhodné barvy trajektorií“ je možné docílit odlišných barev trajektorií jednotlivých objektů, defaultně je totiž pro všechny trajektorie nastavena modrá barva. Uložení nastavení je možné pomocí tlačítka „Potvrdit“. V případě nastavení délky zobrazované trajektorie na hodnotu „0“ je docíleno vypnutí trasování.

**Technické informace:** Trasovací pole je deklarováno pro desetinná čísla (souřadnice středů  $X$  a  $Y$  jednotlivých objektů) je použit typ proměnné Single. To zapříčiňuje poměrně vysoké paměťové nároky na udržení velkého (nastavené vysoké  $n$  u délky zobrazované trajektorie) počtu souřadnic. Typ single u 2 hodnot souřadnic ( $X$ ,  $Y$ ) zabírá v paměti 4 byty (v 32 bitové verzi), což při 15 detekovaných objektech a velikosti trasovacího pole nastavené na 100 (snímků) dává celkem v paměti  $15 \times 100 \times 4 \times 2 = 12000$  kB, proto je maximální možné nastavení délky trajektorie omezeno na 200 snímků (tzv. „framů“).

#### 4.4.8 Nastavení detekce

Tento formulář s názvem „Nastavení detekce“ slouží k nastavení hodnot proměnných, které uchovávají počet skutečných objektů, které by se v regionu s aplikovanou maskou měly nacházet. Uživatel může zadat i nepřesný či přibližný počet objektů, nastavení je ohlížáno a detekce proběhne korektně (viz. Technické informace). Je možné nastavit počet objektů pouze u těch „virtuálních misek“ ve kterých je „vložen“ region, proto jsou na obr. č. 10 některé z misek programově nastaveny na „disabled“

Obr. č. 10 Formulář „Nastavení detekce“ programu DROSANA

Dále umožňuje nastavení tzv. „Maximálního skoku“, což je hodnota (px) která je potřebná pro eliminaci přeskakujících much (vinou špatné detekce či přehozením identit much), které je nejvíce patrné při analýze much v trubičkách.

**Technické informace:** Detekovaný počet objektů nemusí vždy korespondovat s předpokládaným nastaveným reálným počtem objektů, proto jsou na pozadí nastavení jednotlivých reálných počtů much vytvořeny uměle větší hodnoty (defaultně jde o hodnoty o 5 větší než reálné počty much), tím se předejde problémům s detekcí nestandardně detekovaných snímků (rozpad objektů při prahování apod.), algoritmus si pak sám ohlížá identity jednotlivých much, čili „umělé“ vyšší nastavení hodnot reálného počtu much neovlivní analýzu, pouze funguje jako jakási robustní rezerva při vzniku problému.

#### 4.4.9 Nastavení výstupu

Formulář s názvem „Nastavení výstupu“ slouží převážně k nastavení výstupní složky projektu, výstupních parametrů dat (které snímky ukládat), rozsah zpracování dat (uživatelský výběr startovního a konečného snímku analýzy s krokem zpracování). Dále umožňuje tento formulář nastavit tzv. „Průměrování DAT“. Defaultní hodnota, která je zde nastavena (5) určuje jak velké časové (v jednotkách sekund) okno dat rychlosti, bude průměrováno na jedinou hodnotu, čili na průměrnou rychlost za tento časový okamžik. Na výběr je také několik metod kalkulace tohoto průměru rychlosti (KLUZNÝ PRŮMĚR = časová okna se překrývají, PRŮMĚR = časová okna spolu sousedí, MINIMUM = časová spolu sousedí, MAXIMUM = časová okna spolu sousedí, MEDIAN = časová okna spolu sousedí).

Obr. č. 11 Formulář "Nastavení výstupu" programu DROSANA

**Postup:** Uživatel ihned po načtení formuláře může zvolit pomocí tlačítka pro volbu složky ... svou výstupní složku. Název složky bude vygenerován složením prefixu a názvu videa (o prefixu více v Technických informacích). Dále uživatel zvolí rozsah zpracování („Zpracovani\_od“ a „Zpracovani\_do s příslušným „krokem zpracování“), při editaci těchto položek je možné sledovat přepočet na reálný čas snímku ve videu vždy pod příslušným ovladačem. Pořadové číslo posledního snímku videa, čili nastavení proměnné „Zpracovani\_od“, je možné vyčíst z dolní lišty hlavního formuláře. Nastavení a prvotní tvorba nové složky se nastaví po kliknutí na tlačítko „Potvrdit“

**Technické informace:** Název projektové složky je generován pomocí dvou proměnných. První je aktuální čas a datum. Generace je prováděna ve formátu „hh-mm-ss\_DD\_MM\_YYYY“, druhá část názvu je tvořena fyzickým názvem videa (bez

přípony). Takto vygenerovaný název poskytuje stoprocentní jistotu, že se na pevném disku nebudou nacházet žádná duplicitní data. „Průměrování dat“ a „Metoda“ jsou závislé na programu MS Excel. Metodou se rozumí použitá funkce v tabulce XLS, do které jsou ukládány naměřené hodnoty. Průměrování dat je pak pouze počet vteřin, které budou tvořit časové okno k průměrování, čili parametr funkce MS Excel pro analýzu výsledných dat.

#### 4.4.10 Nastavení externího podnětu

Nastavení externího podnětu, čili hodnoty počátečního a konečného snímku videa, ve kterém dochází v případě mého měření k prudkému impulzovitému pouštění vzduchu do prostor trubiček (tzv. „puffání“) s délkou trvání 200 ms na pulz, přičemž počet opakování tohoto puštění je možné navolit na externím zásobovači vzduch (2x, 4x, 6x, 8x), délka mezi jednotlivými „puffy“ činí 1 s (je jí možné nastavit). O aktivaci „puffání“ informuje indikační LED dioda, která svítí po celý průběh pulzu, tzn. v případě kdy je otevřen ventil vzduchu.

Samotné nastavení počátečního a konečného snímku je důležité pro eliminaci vysokých „peaků“ kalkulovaných rychlosti. V případě pulzu je totiž většina much unášena proudem vzduchu k jednomu z okrajů trubičky ve shluku, v tomto případě je algoritmus i samotná obrazová analýza software DROSANA bezmocná a software detekuje pouze rozeznatelné objekty (vše je postaveno na předpokladu, že se všechny objekty pohybují stejnou rychlostí díky pulzu vzduchu, který je pro celou trubičku stejný). Rozsah podnětu vymezený proměnnými („první puff“ a „poslední puff“) určuje tedy, že veškeré „peaky“ vzniklé mimo tento rozsah (před a po „puffání“) vyšší než jsou maximální rychlosti objektu („peak“) nastavené pomocí maximálního skoku (viz. Nastavení detekce“ v kapitole 4.4.8) budou přeskočeny a nezapočítávány do kluzných průměrů.

Nastavení je možné provést přímo z hlavního formuláře programu DROSANA jak je vidět na obr. č. 12. Nastavení se nachází v dolní části formuláře a je reprezentováno barevnými položkami „První puff“ a „Poslední puff“.

**Postup:** Software umožňuje pohybovat se po jednotlivých snímcích otevřeného videa několika způsoby, hrubě pomocí pohyblivého jezdce přímo pod načteným videem, jemně po jednotlivých snímcích pomocí tlačítek „+“ a „-“, nebo zadáním hodnoty snímku, na který má program přejít přímo do kolonky „Skočit na snímek“. Tento pohyb umožňuje uživateli nalézt ty snímky videa, na kterých je zobrazena svítící indikační LED dioda, která signalizuje aktivní externí podnět (aktivní „puffání“) a nastavit začátek aktivity LED diody do příslušných kolonek „Start puffání“ a „Konec puffání“.



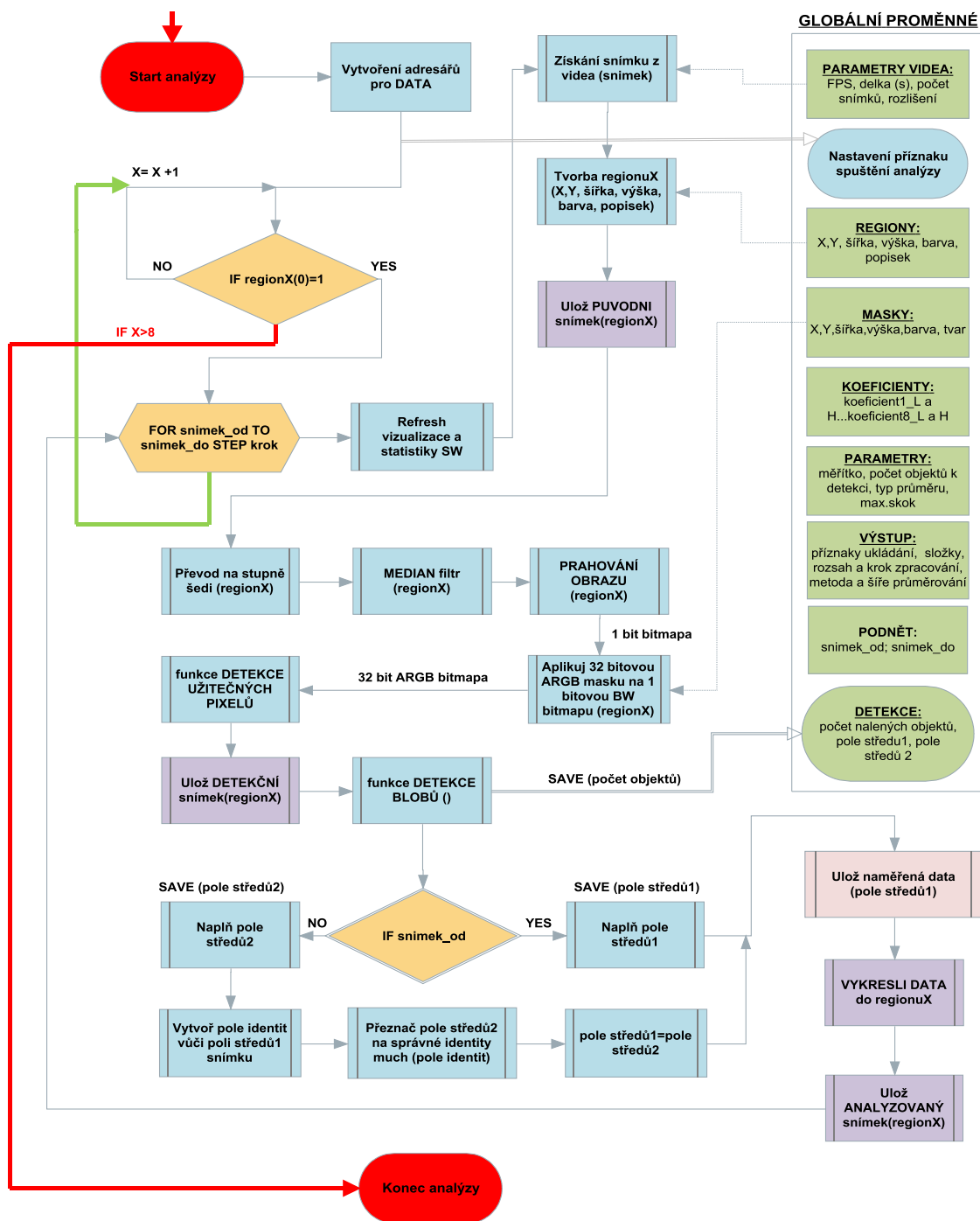
Obr. č. 12 Nastavení externího podnětu na hlavním formuláři programu DROSANA

## 4.5 Návrh zdrojového kódu

Zdrojový kód programu DROSANA jsem vyvíjel něco přes rok a prošel za tu dobu mnoha obměnami. Základní filozofie programu je zřejmá z obr. č. 13, jde především o ukládání uživatelských přednastavení (informace o uživatelských regionech, maskách, koeficientech prahování atd.) do bloku globálních proměnných (toto řeší uživatelské rozhraní popsáno v kapitole 4.4). Z těchto globálních proměnných, které jsou přístupné z jakýchkoliv míst zdrojového kódu jak pro čtení tak i zápis, pak samotná automatická analýza čte hodnoty, které používá ke svému chodu. Samotný program je tvořen modulově, tzn., že každý modul se specifickým zaměřením obsahuje funkce a procedury podobného účelu (např. úprava obrazu, práce se složkami, obrazové filtry apod.). Tyto moduly je pak možné externě použít i v jiných aplikacích.

Na obr. č. 13 je znázorněno obecné jádro programu DROSANA. Start analýzy může začít ve chvíli, kdy jsou všechna uživatelská přednastavení hotova, tzn. uživatel prošel všemi formuláři grafického rozhraní a uložil své nastavení, teprve poté je na hlavním formuláři programově povoleno („enabled“) tlačítko „ZPRACOVAT“.

Zdrojový kód bude psán modulově (MODUL = blok obsahující proměnné, procedury, funkce zabývající se určitou oblastí zpracování). Modulově jsou také zpracovány veškeré globálně dostupné proměnné, před jejichž deklarací předchází příznak „PUBLIC“, který zajistí dostupnost těchto proměnných napříč všemi ostatními moduly a napříč celým zdrojovým kódem.



Obr. č. 13 Vývojový diagram jádra programu DROSANA

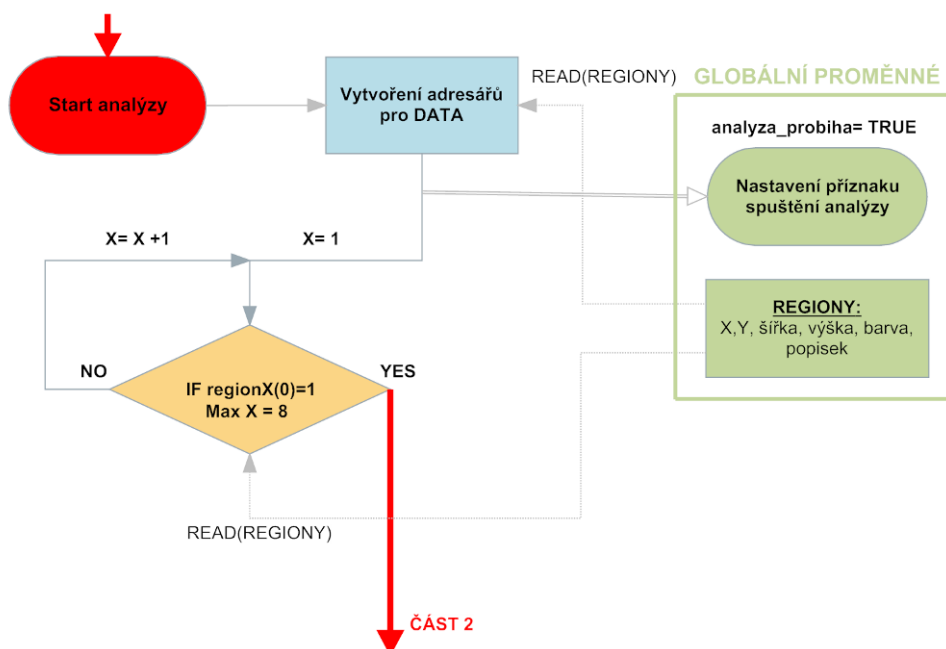
Popis zdrojového kódu jsem rozdělil na pět částí. **ČÁST I.** bude obsahovat popis procedur a funkcí pracujících před samotnou hlavní smyčkou analýzy starající se o vytvoření datových adresářů, přípravu souborů a nastavení příznaků indikující start analýzy. **ČÁST II.** se bude zabývat popisem procedur a funkcí použitých pro „vznik“ obrazu (získání snímku z videa a následnou tvorbou výřezů ze snímku videa, čili tvorbě regionů). **ČÁSTI III.** se bude zabývat předúpravou obrazu (převod do stupňů šedi, medián filtr, prahování obrazu, aplikace masky) v hlavní smyčce analýzy programu DROSANA. **ČÁST IV.** Bude obsahovat popis funkce algoritmu, který se stará

o samotnou detekci objektů v předpřipraveném obraze. K úplnosti bude popsána ČÁST V. která obhospodařuje svými funkcemi a procedurami udržování identit objektů a samotné ukládání výsledných dat. Tímto popisem jsem virtuálně rozdělil jádro programu DROSANA na 5 hlavní části.

## 4.6 Popis částí zdrojového kódu

### 1) ČÁST I.

Na obr. č. 14 se nachází první část hlavního vývojového diagramu, která zodpovídá hierarchicky za nastavení inicializačních příznaků analýzy, za tvorbu datových adresářů a kontrolu uživatelského nastavení.



Obr. č. 14 Část I. vývojového diagramu programu DROSANA

Vytvoření a kontrola existence adresářů je zajištěno pomocí platformy .NET Framework respektive jejich statických metody *CreateDirectory()* a *Exists()* ve třídě *Directory*. První jmenovaná slouží pro tvorbu a přesun adresářů a podadresářů a druhá jmenovaná pro kontrolu existence adresáře [2]. Modul software DROSANA s názvem „Prostredi.vb“ obsahuje proceduru „*prvotni\_vytvoreni\_adresar\_u\_pro\_analyzu()*“, která je postavená právě na těchto užitečných funkcích pro tvorbu a kontrolu adresářů. Následující ukázka zdrojového kódu ukazuje příklad použití této procedury :

```
'Implementace jmenných názvů
Imports System
Imports System.IO
...
Dim cesta_k_adresari As String
```

```
cesta_k_adresari = "C:\PUVODNI_SNIMKY\Miska2
'Vytvoří se adresář, do této funkce vstupuje proměnná typu String
obsahující cestu k adresáři
Directory.CreateDirectory(cesta_k_adresari)
...
Dim existuje_adresar As Boolean
'Zjistí zda-li existuje adresář, do této funkce vstupuje proměnná typu
String obsahující cestu k adresáři (výstupní proměnná je typu boolean)
existuje_adresar=Directory.Exists(cesta_k_adresari)
```

Procedura popsaná výše vytvoří pouze složky a nadřazený projektový adresář (pouze v případě jsou-li potřeba, informace o potřebě tvorby adresáře jsou čteny na základě použití regionu ze sekce Regiony v GLOBÁLNÍCH PROMĚNNÝCH), samotný obsah adresářů a podadresářů je tvořen později. Samotná struktura projektového adresáře, jež je pojmenován pomocí prefixu a názvu videa (viz. kapitola 4.4.9) Nastavení výstupu) se skládá z podadresářů **NAMERENA\_DATA** (obsahují jednotlivé podadresáře s názvy „virtuálních misek, pokud není uveden popis u regionů používají se defaultní názvy „Miska1“, „Miska2“ ..., které obsahují soubory s daty „namerena\_data.xls“ a „data\_rychlosti.xls“), **VYNECHANE\_SNIMKY** (obsahující podadresáře dle názvů „virtuálních“ misek, které obsahují vždy textový soubor se seznamem přeskočených snímků), **PUVODNI\_SNIMKY** (obsahuje podadresáře dle názvů „virtuálních“ misek, které obsahují uložené bitmapy výřezu-regiony snímků z videa), **DETEKCNÍ\_SNIMKY** (obsahuje podadresáře dle názvů „virtuálních“ misek, které obsahují uložené detekční snímky – detekovány pouze pixely objektů videa), **ANALYZOVANE\_SNIMKY** (obsahuje podadresáře nazvané dle „virtuální“ misek obsahující kompletně analyzované regiony-bitmapové obrázky ze snímků videa)

Po Tvorbě adresářů je **nastaven příznak „analiza\_probiha“** na hodnotu TRUE pro následné znemožnění posunu videa z grafického rozhraní programu DROSANA. Programové ovládání krokování komponenty zobrazující video totiž automaticky posouvá svůj jezdec a při uživatelském zásahu v průběhu analýzy by mohlo dojít k pádu programu.

Dalším krokem již začíná jeden z 8 rozhodovacích bloků, kde je testována 0tá hodnota v poli „RegionX\_rozmary()“, kde X je číslo regionu (nabývá hodnot od 1 až 8). Samotné pole, například pro region1 je nazváno „Region1\_rozmary()“, je deklarováno jako 5 rozměrné (včetně 0tého rozměru) pole obsahující proměnné typu Integer. Význam jednotlivých položek pole je uveden v následující tabulce 1:

**Tabulka 1 Význam indexů u pole regionX\_rozmary()**

Položka pole	Funkce
regionX_rozmary(0)	Indikuje zda-li je region použit
regionX_rozmary(1)	X-ová souřadnice regionu
regionX_rozmary(2)	Y-ová souřadnice regionu
regionX_rozmary(3)	šířka regionu
regionX_rozmary(4)	výška regionu



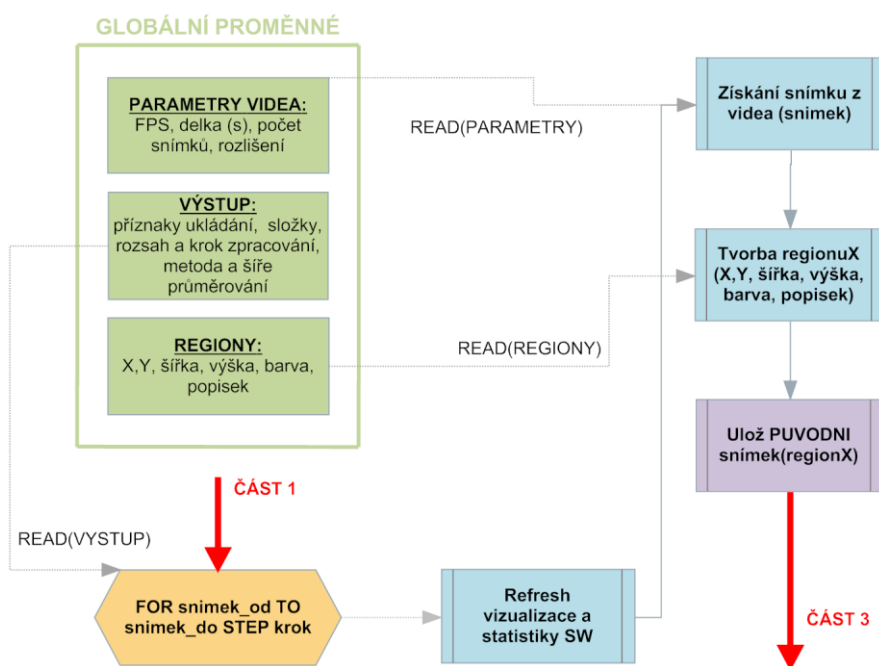
Všechny hodnoty všech 8mi polí s indexem 0 jsou defaultně nastaveny na hodnotu „0“, vybral-li uživatel nějaký region, je v jeho příslušném poli tato hodnota změněna na „1“, což je také cíl testování rozhodovacího bloku, který následuje za nastavením, program tak pozná, zda-li se má pokračovat v analýze tohoto regionu či nikoliv, pokud je tedy hodnota 0vá, je rozhodovací blok přeskočen a hodnota X je navýšena o hodnotu „1“ a rozhodování pokračuje do té doby než je X=8 (defaultně je X=1).

## 2) ČÁST II.

Na obr. č. 15 se nachází druhá část hlavního vývojového diagramu, která zodpovídá hierarchicky za postupné nalezení snímku ve videu, který je následně zkopírován a použit jako zdroj pro uživatelské výřezy které vznikají z této bitmapy (vznikají regiony). Volitelnou větví této části zdrojového kódu je i uložení jednotlivých regionů (výřezů z původní bitmapy, respektive snímku videa).

Po otestování, zda-li není region prázdný (pole RegionX\_rozmary(0)=1), viz předchozí programová ČÁST I., je použit **cyklus FOR s pevným počtem opakování**. Parametry, které vstupují do tohoto cyklu, jsou načteny z GLOBÁLNÍCH PROMĚNNÝCH, přesněji pak ze sekce Výstup (rozsah\_od, rozsah\_do, krok) jež jsou tvořeny proměnnými typu Integer. Program tak pozná, od a do jakého snímku a s jakým krokem (přeskakováním snímků) videa má provádět analýzu.

Před samotnou analýzou je nutné **nastavit grafické rozhraní a statistiku vizualizace průběhu analýzy** (nastavuje se počet analyzovaných snímků, počet přeskočených snímků a spouští se časomíra analýzy a dopočítává se dodatečná statistika související s průběhem analýzy).



Obr. č. 15 ČÁST II. Vývojového diagramu programu DROSANA

V cyklu s pevným počtem opakování (popsaném výše) je nejprve získán snímek (frame) z načteného videa. To je provedeno pomocí funkce s názvem „*Ziskej\_snimek\_z\_vidoa()*“. Tato funkce, k nalezení v modulu s názvem „Prostredi.vb“, jejíž vstupní parametr je číslo snímku, který je potřeba z videa získat, vrací snímek v podobě bitmapy. Základním stavebním kamenem funkce je ovládání **komponenty AxMCIWndX**, která je umístěna na hlavním panelu programu. Poskytuje metody ke krokování videa a samotnému **zkopírování aktuálního snímku videa do schránky** operačního systému. Funkce je popsána následujícím zdrojovým kódem:

```
Function ziskej_snimek_z_vidoa(ByVal snimek As Integer) As Bitmap

    'Deklarace promenné pro výstupní bitmapu
    Dim snimek_z_vidoa As Bitmap
    'vymazání schránky
    Clipboard.Clear()
    'posun videa
    frm_hlavni.AxMCIWnd1.Position = cislo_snimku
    'příkaz pro komponentu
    frm_hlavni.AxMCIWnd1.Command = "stop"
    'vrátí činnost operačnímu systému
    System.Windows.Forms.Application.DoEvents()
    'zkopírování aktuálního snímku videa do schránky
    frm_hlavni.AxMCIWnd1.EditCopy()

    'jestliže již obsahuje schránka zkopírovaný snímek
    If Clipboard.ContainsImage() Then
        'zkopíruj obrázek do proměnné „snimek_z_vidoa“
        snimek_z_vidoa = Clipboard.GetImage()
        'vyčisti schránku
        Clipboard.Clear()
    Else
        'jestliže ne, předej činnost systému a čekej na obrázek
        Do Until Clipboard.ContainsImage()
            System.Windows.Forms.Application.DoEvents()
        Loop
        'zkopíruj obrázek do proměnné „snimek_z_vidoa“
        snimek_z_vidoa = Clipboard.GetImage()
        'vyčisti schránku
        Clipboard.Clear()
    End If

    'funkce vrátí proměnnou obsahující požadovaný snímek videa
    Return snimek_z_vidoa

End Function
```

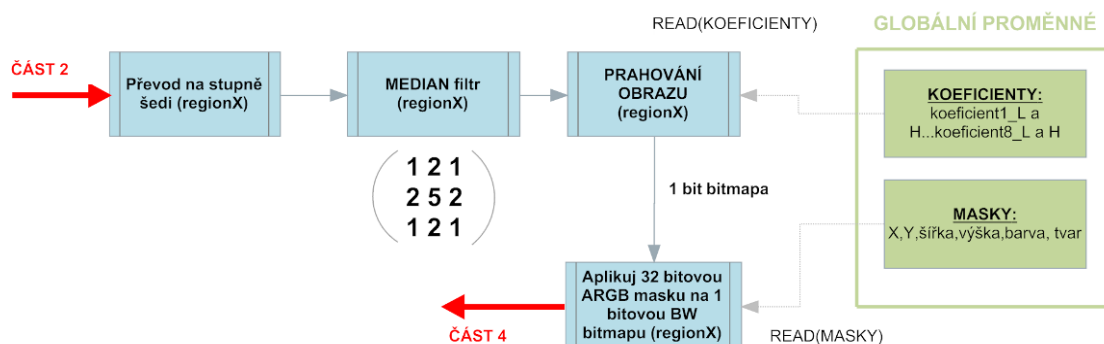
Je-li získán snímek z videa, algoritmus se z něj v dalším kroku snaží vyříznout tu část, která je pro uživatele zajímavá (vytvoří ze snímku menší bitmapu, která obsahuje pouze ten děj, který uživatel potřebuje analyzovat, ostatní je pak oříznuto). Princip této činnosti je jednoduchý, algoritmus si přečte z přednastavených GLOBÁLNÍCH

PROMĚNNÝCH sekci Regiony, dle které nejprve vytvoří bitmapu vhodné velikosti, poté dle nastavení regionu zkopíruje tu část obrazu z původního snímku videa, jež je dána parametry regionu, do nové bitmapy. K tomuto účelu slouží funkce „*Vrat\_vyrez()*“ v modulu s názvem „Obrazky.vb“. Vstupní parametrem této funkce je bitmapa (snímek videa) a číslo regionu (typ Integer), výstupem je bitmapa obsahující výřez z obrázku.

Algoritmus dále pokračuje volitelným (je-li nastaven příznak ukládání původních snímku) uložením regionu do příslušné složky projektového adresáře („PUVODNI\_SNIMKY“) jako obrázek ve formátu \*.png (formát obrázku je programově volitelný).

### 3) ČÁST III.

Tato část vývojového diagramu má na starosti předúpravu obrazu pro samotnou obrazovou analýzu, tzn. převod regionu (bitmapy) do odstínů šedi, dále pak aplikace filtru medián (která vyhladí šum v obraze). Algoritmus dále pokračuje prahování obrazu s následnou aplikací barevné masky. Celý proces a hierarchie této části je možné vidět na obr. č. 16.



Obr. č. 16 Část III. Vývojového programu DROSANA

Převod barevného obrazu (regionu) na šedotónový obraz je výhodný, protože při detekci objektu v obraze není nutné uchovávat informace hned několika barevných složek zároveň (ARGB), ale stačí jedinná (tak aby objekty nezanikly), čímž se ušetří paměť a zrychlí se samotná detekce. Následující zdrojový kód se nachází v modulu „Obrazky.vb“ a činnost převodu barevného obrazu na šedé tóny obstarává funkce „*Stupne\_sedi()*“ jejímž vstupním parametrem je bitmapa (region), výstup tvoří taktéž bitmapa avšak s již převedeným 8 bitovým obrazem ve stupních šedi. Matice colormatrix o velikosti 5x5 je ve formátu RGBAw (R-červený kanál, G-zelený kanál, B-modrý kanál, A-alfa kanál, w-zámek úpravy barev, vždy nastaven na 1). Vhodným nastavením koeficientů této matice, je možné dosáhnout úprav barev pixelů na šedotóny v bitmapě na kterou se matice aplikuje. [3]

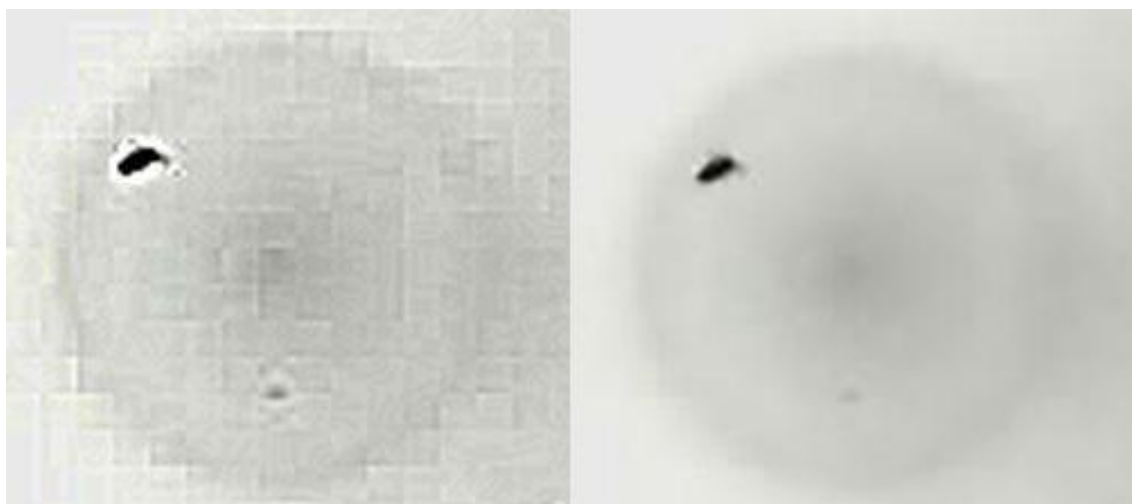
```
Function Stupne_sedi(ByVal vstupni_obrazek As Image) As Image
    'deklarace a inicializace barevné matice
    Dim cm As ColorMatrix
    cm = New ColorMatrix(New Single() () { _
```

```

New Single() {0.33, 0.33, 0.33, 0.0, 0.0}, _
New Single() {0.33, 0.33, 0.33, 0.0, 0.0}, _
New Single() {0.33, 0.33, 0.33, 0.0, 0.0}, _
New Single() {0.0, 0.0, 0.0, 1.0, 0.0}, _
New Single() {0.0, 0.0, 0.0, 0.0, 1.0}})
'výstupem je výstup funkce tvořící nový 8bitový obraz
Return Preved_obraz(vstupni_obrazek, cm)
End Function

```

Po konverzi obrazu regionu do šedého tónu je potřeba zajistit, aby ve vstupním obrazu detektoru nebyl šum (mohl by způsobit „rozpad“ objektů, či komplikovat detekci). Toto odstranění šumu se dle [4] provádí mírným „rozmazáním“ obrazu. Algoritmů pro rozmazání obrazu je celá řada, já jsem se rozhodl pro ten nejjednodušší a nejčastější používaný a to Median Filtr (odstraní šum, ale zachová strukturu objektů). Tento jednoduchý filtr pracuje tak, že každý pixel obrazu nahradí mediánem vypočteným z okolí tohoto pixelu (tzv. jádra). Program DROSANA používá jako okolí jádra sousedních 8 pixelů protože je nutné zachovat co nejpřesnější detaily hran objektů (**LHR – levý horní roh, HR – horní roh, PHR – pravý horní roh, PPR – prostřední pravý roh, DPR – dolní pravý roh, DR - dolní roh, DLR – dolní levý roh, PLR – prostřední levý roh**). U těchto sousedních pixelů jádra filtr zjišťuje hodnotu červené složky (všechny barevné složky RGB mají stejnou hodnotu, pixely se liší pouze hodnotou jasu, jde přeci o obraz v 8 bitových stupních šedi), která je následně vynásobena příslušnou hodnotou v konvulační matici. Konvulační matice (defaultní hodnoty matice pro program DROSANA jsou zobrazeny na obr. 16), velikosti 3 x 3, obsahuje hodnoty, které vyjadřují míru či důležitost jakou se pixely jednotlivých sousedů jádra podílejí na výsledném mediánu.



Obr. č. 17 Vlevo obrázek s patrným šumem; vpravo obrázek po aplikaci MEDIAN filtru

V tuto chvíli má postupující algoritmus kompletně připraven obraz k závěrečné a nejdůležitější části úpravy obrazu, a to k prahování, či jinými slovy k „segmentaci“, která zajistí oddělení pozadí (osvětlená miska či trubička) a popředí (objekty-mouchy). Segmentace pracuje na následujícím principu. Postupně se prochází každý pixel obrazu

a rozhoduje se, zda-li jeho hodnota (jasu, či jedné z barevných složek) je větší než prahovací hodnota, pokud ano stane se pixel černým, pokud je menší stane se pixel bílým.

V mém programu jsem použil jednoduchý jasový typ segmentace, se stejným prahovacím koeficientem pro celý obraz (vhodnější by bylo použití vícera koeficientů při diferencovaném prahování odlišných částí obrazu). Koeficient (horní hranice „koeficientX\_H“) jsem však rozšířil ještě o dodatečný koeficient (dolní hranice, „koeficientX\_L“), kde X označuje číslo regionu, ke kterému prahovací koeficienty přísluší. To software umožní schopnost segmentovat pouze ty pixely, jejichž jas se nachází v rozmezí hodnot vytyčených koeficienty H, L, což by mělo přispět ke zpřesnění vyprahovaného obrazu (zachování co nejvěrnějších hran). Na následujícím zdrojovém kódu je znázorněna funkce „**Vyprahuj()**“, dostupná z modulu „Filtry.vb“. Jejím vstupním parametrem je obrázek (proměnná typu bitmap) jež má být segmentován, dále pak koeficient\_L a koeficient\_H (obě proměnné typu Double), výstupem je obrázek (proměnná typu Bitmap).

```
Function Vyprahuj_obrazek(ByVal zdrojovy_obrazek As Bitmap, ByVal prah_od As Double, ByVal prah_do As Double) As Bitmap
```

```
    'Deklarace proměnných nutných k chodu funkce
    Dim bm As Bitmap
    Dim svetlost As Double = 0
    Dim y As Integer
    Dim x As Integer
    Dim index As Integer

    'Naplní proměnnou „img“ zdrojovou bitmapou
    Dim img As Bitmap = zdrojovy_obrazek
    Dim sirka As Integer = img.Width
    Dim vyska As Integer = img.Heigh

    'Kontrola zda-li je vstupní zdrojovy obrazek 32 bitový
    If img.PixelFormat <> PixelFormat.Format32bppPArgb Then
        'Vytvoří se nová bitmapa „img“ se stejnými rozměry jako
        'měla zdrojová bitmapa
        Dim temp_bitmap As New Bitmap(sirka, vyska,
            PixelFormat.Format32bppPArgb)
        'Na novou bitmapu se zkopíruje zdrojová bitmapa, nyní je
        'však již 32 bitová.
        Dim g As Graphics = Graphics.FromImage(temp_bitmap)
        g.DrawImage(img, New Rectangle(0, 0, sirka, vyska), 0, 0,
            sirka, vyska, GraphicsUnit.Pixel)
        img.Dispose()
        g.Dispose()
        img = temp_bitmap
    End If

    'přístup k bitům obrázku v paměti pomocí Lockbits metody
    Dim bmdo As BitmapData = img.LockBits(New Rectangle(0, 0, sirka,
        vyska), ImageLockMode.ReadOnly, img.PixelFormat)
```

```

'vytvoření bitmapy o barevné hloubce 1 bit
bm = New Bitmap(sirka, vyska, PixelFormat.Format1bppIndexed)
Dim bm_sirka As Integer = bm.Width
Dim bm_vyska As Integer = bm.Height

'přístup k bitům obrázku v paměti pomocí Lockbits metody
Dim bmdn As BitmapData = bm.LockBits(New Rectangle(0, 0,
bm_sirka, bm_vyska), ImageLockMode.ReadWrite,
PixelFormat.Format1bppIndexed)

'projdi všemi pixely o souřadnicích X,Y
For y = 0 To img.Height - 1
    For x = 0 To img.Width - 1

        'vygenerování adresy prvního barevného pixelu (modrý B),
        'index+1= adresa (zelená-G), index+2= adresa (červená-R)
        index = y * bmdo.Stride + x * 4

        'nalezení světlosti pixelu o souřadnicích X,Y barevných
        'složek R,G,B v paměti
        svetlost = (Color.FromArgb(Marshal.ReadByte(bmdo.Scan0,
        index + 2), Marshal.ReadByte(bmdo.Scan0, index + 1),
        Marshal.ReadByte(bmdo.Scan0, index)).GetBrightness())

        'Zkontroluje světlost(jas) pixelu, je-li mimo hranici bude
        'pixel nastaven na bílý (1.00), jinak na defaultně nataven
        'černý (0.00, tohoto triku se využívá i při rozhodování.
        If (svetlost < prah_od) or (svetlost > prah_do) Then
            'nastavovací procedura pro bílý pixel.
            Nastavit_pixel_prahovani(x, y, bmdn, True)
        End If

    Next x
Next y

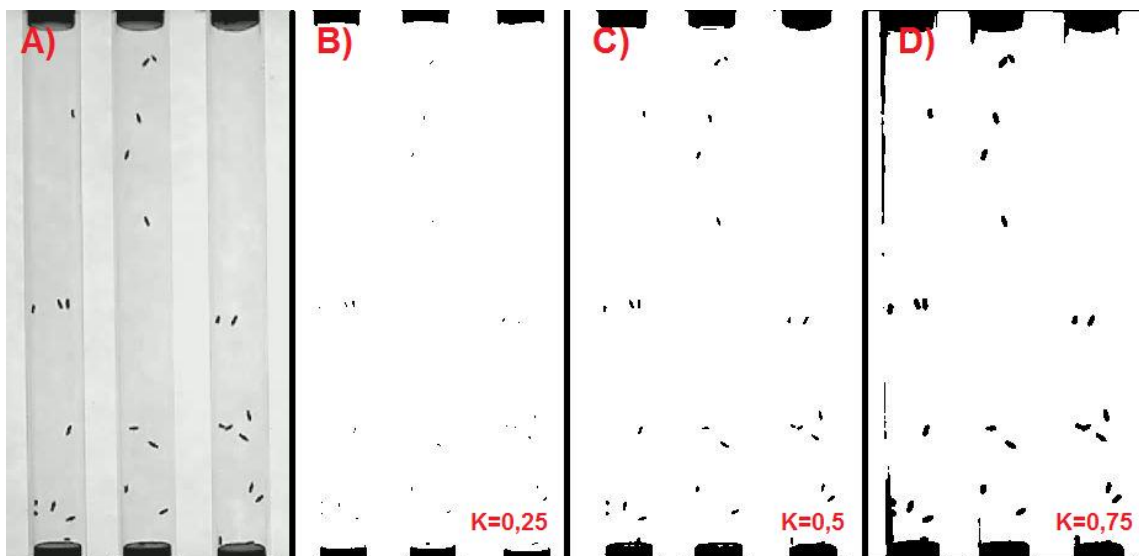
'uvolnění přístupu k bitmapám v paměti
bm.UnlockBits(bmdn)
img.UnlockBits(bmdo)
'funkce vrátí výslednou vyprahovanou bitmapu
Return bm

End Function

```

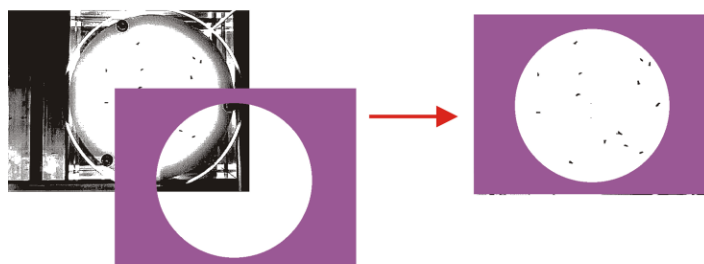
Jak je vidět ze zdrojového kódu, přístup k datům obrázku není řešen přes pomalé nativní metody *GetPixel()* a *SetPixel()*, které používá nativně Visual Basic .NET, ale je řešen přes metodu *LockBits* a *UnlockBits* třídy *Marshal*, která umožňuje VB.NET jinak zakázaný přístup do paměti (jazyk totiž nepodporuje oficiálně fungující pointery či speciální klíčová slova). Samotný zápis barvy pixelu (černá, či bílá) je řešen v podfunkci „*Nastavit\_pixel\_prahovani()*“ jejímž vstupem je souřadnice X a Y pixelu, dále pak „*bmdn*“ což jsou bitmapová data a **příznak** typu Boolean, který informuje funkci o logice, kterou má pixely obarvovat (TRUE=černá, FALSE=bílá).

Filozofie použití třídy Marshal je popsána ve zdroji [5] odkud jsem také čerpal, výše zmíněný kód je velmi silnou modifikací použití této třídy na specifický účel. Zdroj mi pomohl v pochopení principu přístupu VB.NET do paměti. Segmentace obrazu hraje v případě obrazové analýzy klíčovou roli, na obr. č. 18 je znázorněn vliv hodnoty horního limitu prahovacího koeficientu na prahovací účinek (dolní limit nastaven na „0“).



Obr. č. 18 A) Původní obrázek na vstupu funkce "Vyprahuj\_obrazek"; B) Výsledný segmentovaný obraz ( $K_L=0$ ,  $K_H=0,25$ ); C) Výsledný segmentovaný obraz ( $K_L=0$ ,  $K_H=0,5$ ); D) Výsledný segmentovaný obraz ( $K_L=0$ ,  $K_H=0,75$ );

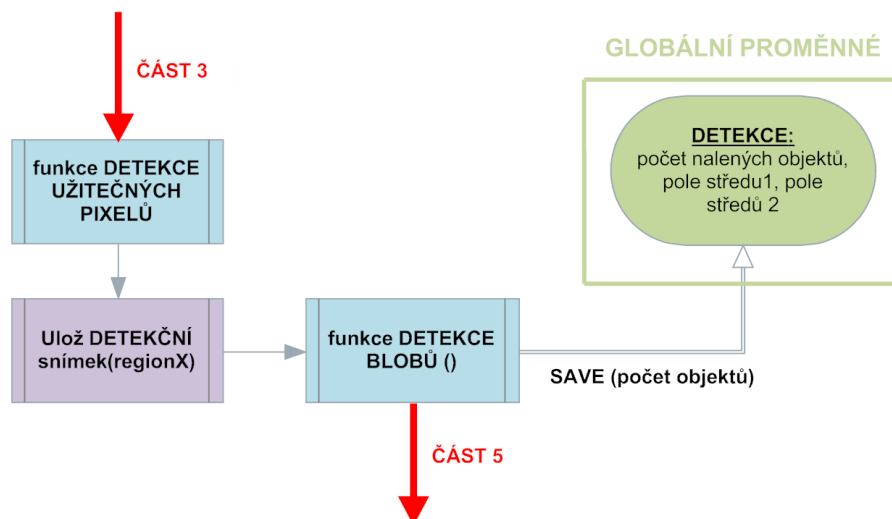
Je-li obraz již segmentován, přistupuje algoritmus k dalšímu kroku a to je překrytí segmentovaného obrazu (1 bitu) pomocnou barevnou maskou (32 ARGB). Tato maska dokoriguje jemné okraje. Například na obr. č. 18 D) je vidět, že trubičky vykazují po segmentaci stále okraje svých stěn. Tento neduh lze odstranit právě překrývající maskou. (maska je vlastně barevná bitmapa, která díky svému typu ARGB, umožňuje používat tzv. alfa kanál, tzn. průhlednost, čehož je využíváno). Do bitmapy barevné masky je vykreslen bílý 2D objekt (kruh či obdélník) přičemž jeho bílá barva je nastavena pomocí alfa kanálu na hodnotu „0“ tzn., že bílá barva se stane průhlednou (okolní barva masky zůstane neprůhledná). Filozofii použití překrývající masky zobrazuje následující obr. č. 19:



Obr. č. 19 Princip krycí masky programu DROSANA

Překrytím maskou, čili bitmapou s barevnou hloubkou 32bit typu ARGB se z 1bitového obrazu nesoucího obrazová data stane bitmapa s barevnou hloubkou a typem jako má maska samotná.

#### 4) ČÁST IV.



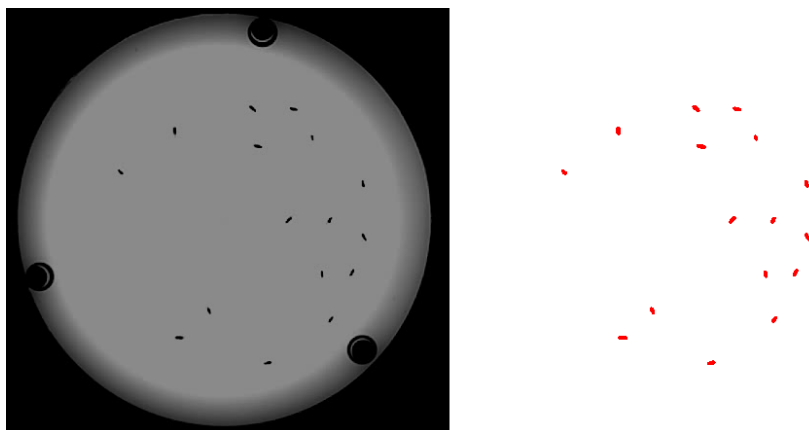
Obr. č. 20 Část IV. Vývojového diagramu DROSANA

Čtvrtá část vývojového diagramu se zabývá samotnou detekcí objektů v obraze. Tato hlavní část s detektory je v software DROSANA řešena oproti konkurenčnímu software mírně nestandardně a to více průchody a bez použití rozdílových snímků (odečet snímku s objekty od snímku bez objektů či častější pro detekci pohybu odečet dvou následujících snímků), avšak toto řešení se jeví jako vhodnější, jelikož odpadá nutnost pořizovat (kalkulovat) snímek pozadí bez objektů (podobně jako v případě konkurenčního software CTRAX) a odečet následujících snímků by dostatečně nepomohl při detekci nepohybujících se objektů.

Princip detekce užitečných pixelů má na starosti funkce „Detekce\_much()“ v modulu „Filtry.vb“, jejímž vstupem je výsledná bitmapa s maskou (výstup ČÁSTI III.) a výstupem bitmapa, která je modifikována dle následujícího schématu: pixely které nejsou černé (barevná maska, POZOR! Barva masky nesmí být černá či bílá) jsou vymazány (pixely získají bílou barvu), veškeré ostatní objekty (tzn. objekty v obraze – mouchy) jsou obarveny na červenou barvu, vznikne tak obraz o velikosti původní bitmapy (regionu), který obsahuje pouze červené pixely objektů, které je potřeba identifikovat. (pozadí bitmapy je bílé).

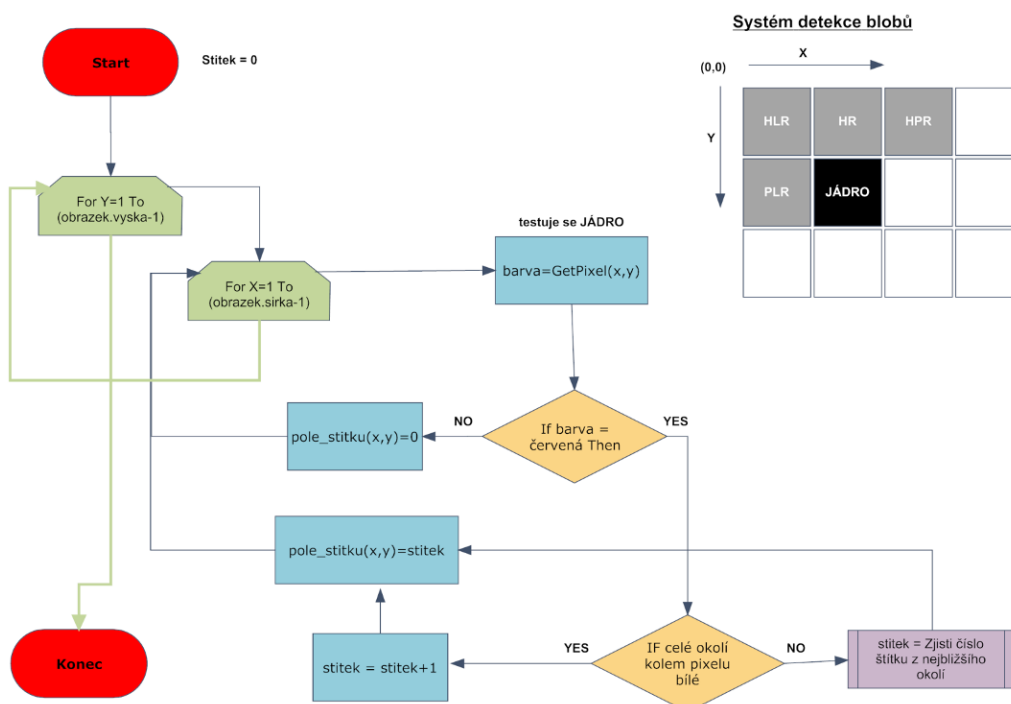
Detekční snímek (příklad zobrazen na obr. č. 21), jak je tato výše popsána bitmapa popsána, je možné volitelně uložit na pevný disk pro následnou kontrolu. Tato možnost je dostupná pouze v betaverzi programu.





Obr. č. 21 Vlevo původní snímek videa; Vpravo detekční snímek programu DROSANA

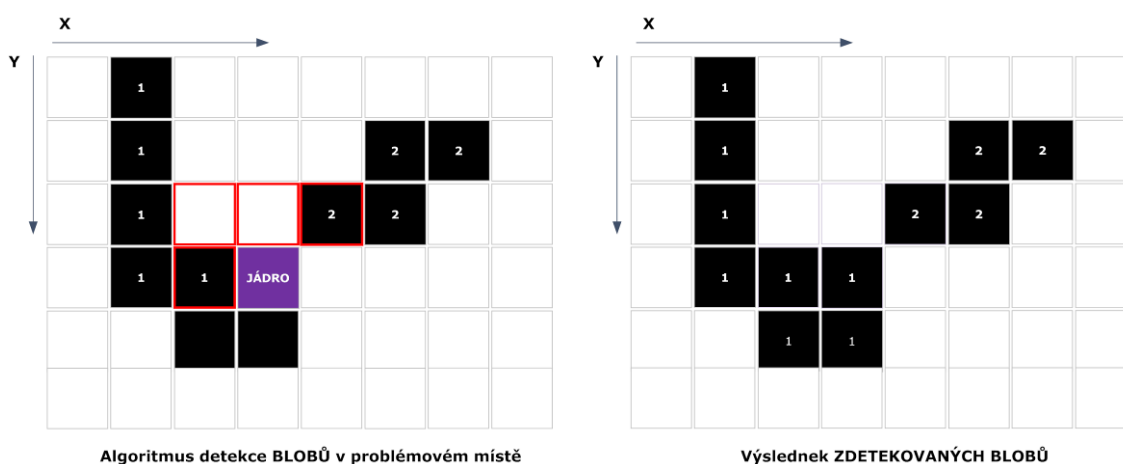
Finálně upravený snímek (vyhlazený - vyprahovaný - překrytý maskou - detekovány jen užitečné pixely) je nyní připraven pro **detekci blobů** (blob = skupina jednotek obrazu tvořící ucelený a jedinečný objekt). Zavoláním funkce „*Detekce\_blob()*“, který se nachází v modulu „*Filtry.vb*“ dojde k detekci blobů v obraze dle následujícího schématu:



Obr. č. 22 Vývojový diagram algoritmu pro detekci BLOBŮ v programu DROSANA

Algoritmus [6] prochází jednotlivé pixely obrazu (tzv. jádra), ty jsou testovány na svoji červenou složku (objekty v obraze totiž jsou obarveny právě na červeně). Nemí-li červená složka = 255 jedná se o bílou barvu, tudíž jde o pozadí. Do pole štítků o daných souřadnicích se uloží číslo štítku s hodnotou „0“, tzn., že jde o pozadí. Je-li červená složka pixelu = 255, jde o pixel, který patří k nějakému objektu, testuje se proto okolí tohoto pixelu (jádra). Je-li okolí bez červených pixelů, jedná se o počáteční pixel

nového nalezeného objektu (nového štítku, proto je tato hodnota zvětšena o „1“), do pole štítků na daných souřadnicích se zapíše hodnota štítku (v případě prvního nového tedy „1“). Je-li v okolí nalezeného jádra nějaký červený pixel (používá se první nalezený), použije se právě jeho číslo štítku k označení jádra (pixely tak k sobě patří). K prohledávání okolí pixelu (jádra) se používá matice zobrazená na obr. č. 22 (zobrazena mimo vývojový diagram). Matici tvoří pixely označeny HLR = horní levý roh, HR = horní roh, HPR = horní pravý roh, PLR = prostřední levý roh. Pixely, které jsou po okrajích obrazu, mají část okolí mimo rozsah pixelů obrazu, jako testovací hodnoty jsou proto používány virtuální pixely s hodnotami štítků „0“ (čili pozadím). Tato detekce blobů je velmi účinná, avšak dochází díky tvaru matice k problému kalkulace více štítků, než odpovídá realitě (blob tvoří pixely s různými štítky), tento problém je znázorněn na obr. č. 23:



Obr. č. 23 Znázornění nekorektní detekce blobů prvním průchodem algoritmu

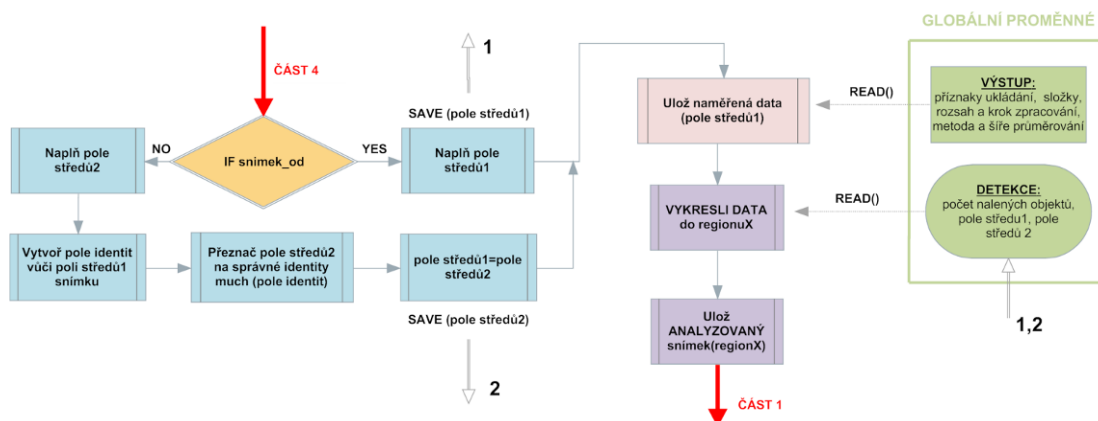
Problém, který je způsoben tvarem matice a používání prvního nalezeného sousedního štítku v okolí je řešen pomocí tzv. korekce štítků. Ta je provedena pomocí dodatečné funkce s názvem „*Korekce()*“ v modulu „*Filtry.vb*“, jejím vstupem je počet nalezených štítků (typ proměnné Byte), výška a šířka obrázku (obě proměnné jsou typu Integer). Výstup pak tvoří proměnná typu Byte, která obsahuje korektní počet již samostatných štítků. To je zajištěno tím, že je opět procházeno pole štítků a detekovány ty dvojice štítků, které se vzájemně dotýkají. Jsou-li takové dvojice štítků nalezeny, je celý blob označen nejmenší hodnotou štítku z dvojice štítků a všechny následující štítky v poli štítků zmenšeny o hodnotu „1“.

Dodatečnou funkcí, která detekci blobů doplňuje, je funkce s názvem „*Korekce\_sumu()*“ v modulu „*Filtry.vb*“, jejím vstupem je počet nalezených štítků po korekci (proměnná typu Byte), výška a šířka obrázku (obě proměnné jsou typu Integer). Výstupem je korigovaný počet štítků. Tato funkce je do programu DROSANA zařazena z důvodu eliminace velmi malých objektů, může se totiž stát, že se v obraze objeví nežádoucí objekt (např. kus žrádla, část těla hmyzu, nečistota apod.), která by detekci navýšila o objekt, byť je tento objekt pro samotnou analýzu nepodstatný. Funkce pracuje tak, že prochází a počítá sumu pixelů jednotlivých štítků, sestaví z nich sestupnou řadu a určí hodnotu MEDIAN. Je-li některá suma pixelů konkrétního štítku

menší než 25 % hodnoty MEDIAN, jsou tyto pixely štitku smazány (přepsány na hodnotu pozadí, čili „0“) a všechny následující štitky v poli štitků sníženy o hodnotu „1“.

## 5) ČÁST V.

Poslední pátá část vývojového diagramu má na starosti systém udržování identit much a ukládání výsledných dat v určitém formátu na předem dané místo.



Obr. č. 24 ČÁST V. Vývojového diagramu programu DROSANA

Po detekci blobů (viz. ČÁST IV.) je v paměti dostupná informace o počtu nalezených objektů (hodnota proměnné „stitek“) a dále číselná mapa umožňující určení kde se který objekt nachází včetně pixelů, které k němu náleží (proměnná „pole\_stitku“).

Algoritmus analýzy programu DROSANA dále pokračuje **výpočtem jednotlivých souřadnic** středů blobů, respektive objektů. Souřadnice středu jsou kalkulovány tak, že jsou u blobu hledány minimální a maximální souřadnice pixelů, které blob tvoří a to jak v ose x tak v ose y. Dle následujících rovnic je pak možné vypočítat středy blob:

$$X_S = (X_{MAX} - X_{MIN})/2 \quad (1)$$

$$Y_S = (Y_{MAX} - Y_{MIN})/2 \quad (2)$$

O kalkulaci středů a naplnění polí středů těmito výpočty (čili středy všech nalezených objektů) se stará funkce s názvem „*Vytvor\_pole\_stredu()*“ jejímž vstupem je hodnota proměnné „stitek“ (typu Byte), čili počet zdetekovaných objektů, dále pak výška a šířka pole štitků (typ proměnných Integer). Návratovou hodnotou je dvourozměrné pole o velikosti („stitku“x2), jež obsahuje souřadnice X, Y (typ proměnné Single) vždy konkrétního objektu. Jde-li o souřadnice detekovaných objektů prvního snímku videa, jsou uloženy do proměnné „pole\_stredu1(,)“, jde-li o souřadnice ostatních snímků videa, jsou souřadnice ukládány do proměnné „pole\_stredu2(,)“. V případě pole středů 1 není potřeba hledat identity much z předchozího snímku, je tedy možné data uložit rovnou (systém ukládání je popsán dále).

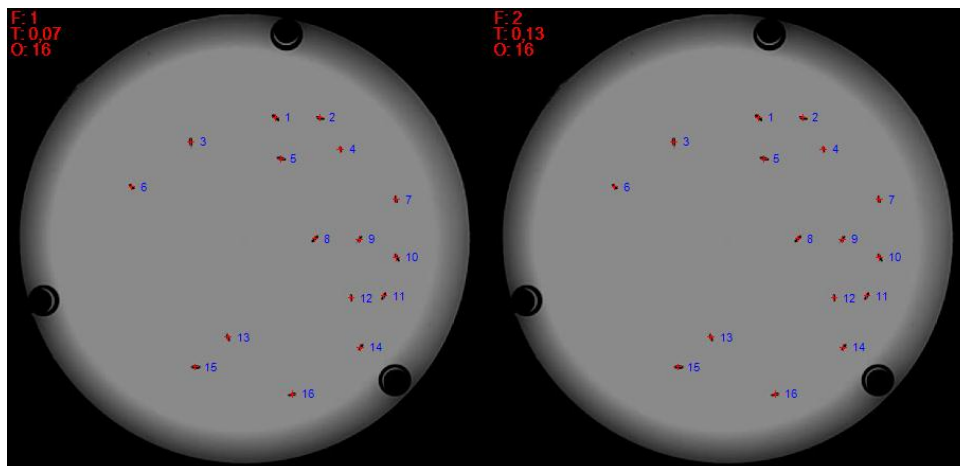
Potřeba udržovat identity much po co nejdelší dobu (optimálně po celou délku videa) je zajištěna pomocí systému hledání nejbližších much v předcházejícím snímku, čili v hledání u dvou po sobě jdoucích snímků (respektive v proměnných „pole\_stredu1(,)“ a „pole\_stredu2(,)“) a používá se pouze v případě, že jde o detekci jiného než prvního snímku videa. Po naplnění proměnné „pole\_stredu2(,)“ je zavolána funkce „Nejblyzi\_mouchy()“, nachází se v modulu „Filtry.vb“, jejímž vstupem jsou obě výše zmíněné pole (položky polí jsou proměnné typu Single), výstupem této funkce je jednorozměrné pole s názvem „POLE IDENIT“. Funkce vygeneruje dvourozměrnou matici vzdáleností mezi všemi objekty navzájem. Matice, jejíž velikost daná velikostmi prvních rozměrů polí „pole\_stredu1“ a „pole\_stredu2“ používá k výpočtu vzdálenosti vztah (3).

$$s = \sqrt{(X_1 - X_0)^2 + (Y_1 - Y_0)^2} \quad (3)$$

(\* souřadnice  $X1, Y1$  jsou brány z pole\_stredu2, souřadnice  $X0, Y0$  pak z pole\_stredu1)

Poté algoritmus pracuje dvoufázově, nejprve přiřazuje-páruje ty objekty, jejichž vzdálenost je „0“, zároveň tyto objekty zamkne před dalším případným duplicitním párováním. Poté je vždy postupně (až do přiřazení-zpárování všech objektů) hledána vždy nejmenší vzdálenost v matici a následně jsou párovány příslušné objekty (zároveň zamykány proti dalšímu párování). Algoritmus končí svou činnost ve stavu, kdy je vytvořen maximální počet párů, který odpovídá situaci. Počet párů je proměnný, v závislosti na množství rozpoznávaných objektů ve dvojici po sobě jdoucích snímků (např. první ze snímků obsahuje 6 objektů, následující 8 objektů, čili je možné vytvořit maximálně 6 párů), platí nepsané pravidlo: **„počet párů, které je možné vytvořit, je určen nejmenší hodnotou detekovaných objektů z dvojice snímků“**. Proměnná „Pole\_stredu2(,)“ je pak na základě POLE IDENITIT přeorganizována (jsou přesunuty vždy dvojice souřadnic na místa, které odpovídají daným identitám objektů). Pro další použití dat z předchozího snímku je nutné při přesunu k analýze nadcházejícího snímku použít „pole\_stredu2(,)“ jako „pole\_stredu1(,)“, čili dojde ke zkopírování dat a případné změně velikosti „pole\_stredu1(,)“ (v případě že se lišili velikosti těchto polí). Samotné identity much je možné volitelně ukládat do složky „ANALYZOVANÉ SNÍMKY“, na obr. č. 25 je zobrazen náhled systému identit objektů v průběhu analýzy.

Systém ukládání dat je realizován do adresáře NAMERENA\_DATA a to přímo do souboru s příponou CSV (název souboru „namerena\_data.csv“). Je tak zajištěna jeho takřka neomezená velikost (v případě diskového souborového systému FAT32 omezení velikosti na cca 4GB).



Obr. č. 25 Systém zobrazování identit jednotlivých objektů programu DROSANA, v horním rohu jsou zobrazeny dodatečné informace o snímku (F = frame (-), T= time (s), O= objects (-))

Princip souborů CSV je takový, že data jsou od sebe oddělena v řádce středníkem „;“, stejné pořadí dat v řádcích pak tvoří sloupec, čili výhodou je tak snadný import do MS Excel, který data sám rozdělí do jednotlivých buněk, data tak tvoří přehlednou tabulku. Data jsou ukládána ve formátu:

frame (-); time (s); objects (-); x1 (px); y1 (px); s1 (px); s1 (mm); v1 (px/s); v1 (mm/s); a1 (mm/s)...

#### Vysvětlivky:

\*Červeně označené položky se již v řadě neopakují

frame	číslo analyzovaného snímku videa
time	časový okamžik snímku ve videu (na základě FPS)
objects	počet detekovaných objektů
x1, y1	souřadnice detekovaného objektu č.1
s1 (px)	vzdálenost, kterou objekt urazil mezi dvěma snímky v px
s1 (mm)	vzdálenost, kterou objekt urazil mezi dvěma snímky v mm
v1 (px/s)	rychlost, kterou vyvinul objekt mezi dvěma snímky v px/s
v1 (mm/s)	rychlost, kterou vyvinul objekt mezi dvěma snímky v mm/s
a1 (mm/s)	zrychlení, kterého objekt dosáhl mezi dvěma snímky v mm/s <sup>2</sup>

Rychlost a zrychlení objektu jsou kalkulovány teprve až při ukládání dat na rozdíl od vzdálenosti, kterou objekt urazí mezi dvěma snímky (ta je brána z již hotové matice vzdáleností, která vznikla při hledání identit much viz. ČÁST V.). Rychlost je počítána na základě rovnice (4), zrychlení pak na základě vztahu (5).

$$v = s/dt \quad (4)$$

$$a = v/dt \quad (5)$$

(\* s je vzdálenost z rovnice (3), v je rychlost z rovnice (4), dt je časový údaj, daný FPS videa, je dán dt=1/FPS)

Jelikož se ukládají analyzované snímky ve formě jednotlivých obrázků (formát PNG), je potřeba pro video s výslednou vizualizací analýzy tyto obrázky spojit dohromady. Pro tuto činnost je společně s programem DROSANA dodán externí

program známý z operačního systému LINUX jménem Mencoder.exe, který pomocí dodaného skriptu dokáže po zkopírování skriptu i programu Mencoder.exe do složky s obrázky po spuštění skriptu spustit sloučení všech snímků do výsledného videa za použití kodeku MPEG-4 (kontejner AVI). Následující skript převede snímky ve formátu PNG ve stejné složce se skriptem na video s výstupním rozlišením 800x600 px o snímkové frekvenci 15 FPS:

```
echo off
mencoder.exe mf://*.png -mf w=800:h=600:fps=15:type=png -ovc lavc -
lavcopts vcodec=mpeg4 -oac copy -o vystup.avi"
```

## Použité zdroje

- [1] VB.NET: Visual Basic .NET FAQ[online]. c2007 [cit. 2011-11-04]. Dostupné na World Wide Web: <<http://www.vbnet.cz/faq.aspx>>
- [2] Microsoft: MSDN .NET Framework [online]. c2012 [cit. 2011-11-04]. Dostupné na World Wide Web: <<http://msdn.microsoft.com/cs-cz/library/54a0at6s.aspx>>
- [3] Microsoft: MSDN .NET Framework 4: Třída Colormatrix [online]. c2011 [cit. 2011-01-04]. Dostupné na World Wide Web: <<http://msdn.microsoft.com/cs-cz/library/system.drawing.imaging.colormatrix.aspx>>
- [4] BURGER, Wilhelm a Mark BURGE. Principles of Digital Image Processing: Fundamental Techniques. London: Springer-Verlag, 2009, Median Filter. ISBN 978-1-84800-190-9.
- [5] BobPowell.net: Using the LockBits method [online]. c2003 [cit. 2011-11-04]. Dostupné na World Wide Web: <<http://www.bobpowell.net/lockingbits.htm>>
- [6] The Lab Book Pages: An online collection of electronics information, Blob detection [online]. c2009 [cit. 2011-11-04]. Dostupné na World Wide Web: <<http://www.labbookpages.co.uk/software/imgProc/blobDetection.html>>
- [7] GÜRTLER, Martin a Pavel KOCICH. Visual Basic .NET: Hotová řešení. První vydání. Brno: CP Books a.s., 2005, 312 s. ISBN 80-251-0367-6. s. 76-85

## 5 POROVNÁNÍ NAMĚŘENÝCH VÝSLEDKŮ

Program DROSANA je určena pro analýzu videí ve formátu AVI a WMV, které obsahují nasnímaný pohyb hmyzu, zejména je pak uzpůsoben pro Octomilky obecné. Analýzu je schopen provést z prostorů misek, kde je schopen zaznamenávat polohu objektů, vzdálenost kterou objekt ze snímku na snímek urazí a dále pak rychlost a zrychlení, které objekt vyvine ze snímku na snímek. Dále pak z prostorů úzkých trubiček, které slouží pro testování reakčních dob a průběhu uklidňování hmyzu po uměle vyvolaném podnětů (krátkodobém volitelně nastavitelném proudu vzduchu). Prvního typu analýzy (misky) je schopen z konkurenčních programů pouze CTRAX. TriKinetics a Pysolo nejsou na takovou analýzu stavěny (neumožňují buď vstup ve formě videa pro analýzu, či nedokážou sledovat více objektů najednou).

Proto jsem provedl test a porovnání výsledných dat obou programů (DROSANA vs CTRAX) na stejném testovacím videu (rozlišení 800x600 px, 15 FPS, délka cca 5 minut) s jedinou mouchou. Oba programy detekovali správný počet snímků z hlavičky souboru (cca 4522 snímků). Hlavní výsledky jsou zobrazeny v přehledné tabulce 1.

**Tabulka 1 Porovnání naměřených dat programy DROSANA a CTRAX**

DATA	Drosana	CTRAX	Odchylka	Jednotky	Odchylka (%)
Celková naměřená vzdálenost	12796,65	12915,82	119,17	px	<b>0,93</b>
Celková naměřená vzdálenost	4265,12	4304,84	39,72	mm	<b>0,93</b>
Průměrná rychlost mouchy	42,45	42,85	0,40	px/s	<b>0,95</b>
Průměrná rychlost mouchy	14,15	14,28	0,13	mm/s	<b>0,95</b>
Průměrné zrychlení mouchy	445,31	213,62	231,62	px/s <sup>2</sup>	<b>52,01</b>
Průměrné zrychlení mouchy	148,42	71,20	76,8	mm/s <sup>2</sup>	<b>51,7</b>

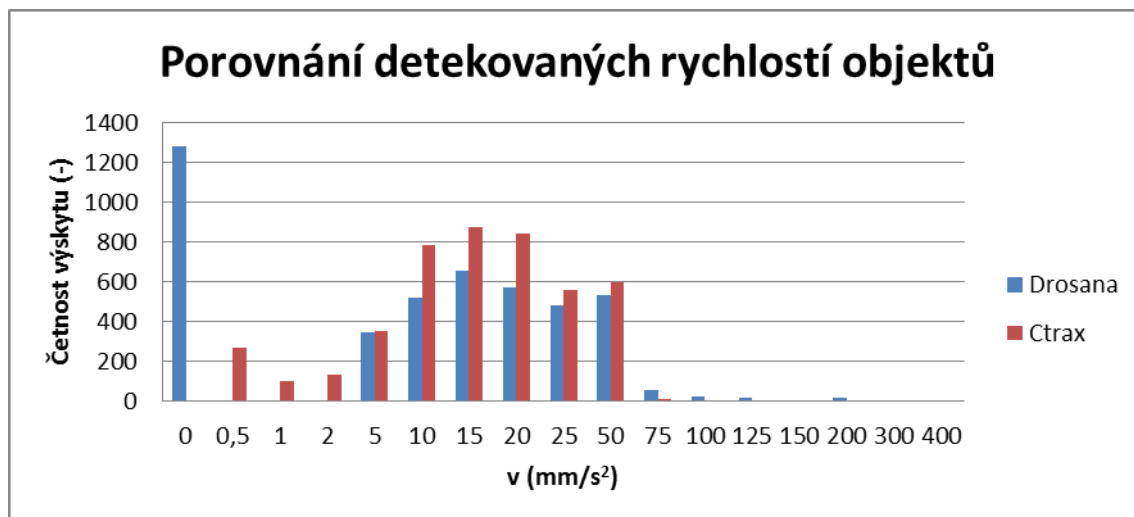
Z tabulky 1 je vidět, že si program můj program DROSANA nevedl oproti programu CTRAX, který je vyvíjen intenzivně autory zhruba již 4 rok vůbec zle. Naměřená vzdálenost se mezi programy liší o 0,93 %, což je způsobeno několika faktory. Software CTRAX používá ke kalkulaci středu objektu těžiště, které je zaokrouhlováno na 4 desetinná místa, byť nejmenší rozlišitelná jednotka obrazu, který program analyzuje je 1 px (dopouští se tak umělého zavádění chyby), software DROSANA používá ke kalkulaci středu objektu jeho velikost v ose x, y, která je následně podělena „2“, získá se tak střed objektu, který je zaokrouhlován na 1 desetinné místo (čili rozlišení tvoří 0,5 px). Přepočtem na jednotky „mm“ je možné zjistit, že odchylka v konkrétním měření na celkové uražené vzdálenosti činí 3,972 cm z celkových cca 4,2 m, což je zanedbatelný rozdíl.

Průměrná rychlost mouchy je u obou programů kalkulována stejnou metodou, avšak díky odlišně detekovaným vzdálenostem, které moucha urazila ze snímku na snímek (potažmo celkové uražené vzdálenosti) se mírně liší. Program DROSANA umožňuje uživateli kontrolu na FPS videa, je tedy jistí, že je rychlost kalkulována na základě korektních nastavení, zatímco program CTRAX používá ke kalkulaci FPS



videa údaj z hlavičky vstupního souboru videa a uživatel tak nemá kontrolu nad případnými nekorektně přečtenými daty. I přesto odchylka mého programu od programu CTRAX činí 0,95 % což v přepočtu na jednotky „mm“ znamená zanedbatelnou odchylku o 0,13 mm/s z celkové průměrné hodnoty cca 14,1 mm/s.

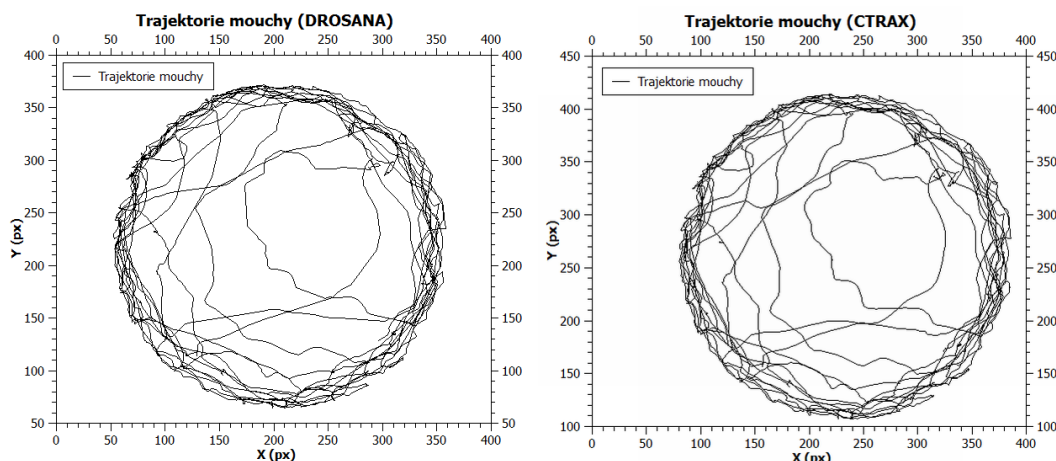
Srovnají-li se jednotlivé statistiky naměřených výsledků od obou programů, je možné zjistit, že oba programy poskytují velmi odlišné výsledky hodnot rychlostí a zrychlení (srovnáno dle četností zobrazených na obr. č. 1), což je možné přisuzovat právě odlišnému způsobu detekce souřadnic středů.



Obr. č. 1 Porovnání četností detekovaných rychlostí programy DROSANA a CTRAX

Díky adaptivnímu prahování (v případě nekorektní detekce při vyprahování jsou snižovány či zvyšovány prahovací koeficienty pro obraz) a samotné kalkulaci rychlostí na 4 desetinná místa, je kalkulace těžiště objektu v programu CTRAX vždy rozdílná. Objekt se v blízkosti jiných objektů (je využito adaptivního prahování) jeví posunutý oproti snímku, kde je ten samý objekt, ale již bez svého okolí, na té samé pozici, s těmi stejnými parametry, se stejným natočením (tvoří jej více či méně pixelů ovlivněné prahovacími koeficienty a také navíc šumem), což je jeden z důvodů proč nedochází u programu CTRAX ke snímání rychlostí s nulovou hodnotou a četností které dosahují se svým programem u nulových rychlostí, jsou rozptýleny u programu CTRAX do nízkých hodnot (0,5 až 2 mm/s).

Při srovnání trajektorií, jak je zobrazeno na obr. č. 2, je možné vidět u trajektorie generované programem CTRAX, offset cca 50 px v obou osách, to je způsobeno rozdílným systémem přístupu obou programů k analýze dat, samotný program CTRAX používá k analýze celý snímek videa, kdežto program DROSANA používá pouze užitečný výřez (region).



Obr. č. 2 Porovnání trajektorií generovaných z naměřených dat programů DROSANA a CTRAX

Díky takřka 4 letému vývoji programu CTRAX s bohatou programátorskou základnou, je však tento software dále, poskytuje díky speciálním skriptům běžících v programu MATLAB zpracování dat, které umožní generaci grafů a výstupů zobrazujících behaviorální chování objektů, statistiku pohybu (včetně úhlu natočení objektů – byť není příliš spolehlivé a je potřeba jej korigovat pomocí dodávaného rozhraní pro MATLAB „FixErrorGUI“, dále pak tzv. heatmap, což je vlastně graf, zobrazující barevnou intenzitou četnost návštěv souřadnic obrazu na, kterých se objekt zdržoval nejvíce) pro každou jednotlivý objekt. Bohužel má tento software poměrně velké problémy s vlastními pořízenými videa obsahující větší počet objektů, velmi snadno dochází při detekci k přeskokům identit, a ke zhroucení analýzy, což u programu DROSANA díky struktuře hrozí jen v extrémních případech.



Obr. č. 3 Porovnání vizualizace naměřených dat programů CTRAX a DROSANA

Porovnáním výsledných naměřených dat programu DROSANA s konkurenčním programem CTRAX jsem získal představu o kladech a záporech svého programu. Díky tomu že můj program DROSANA neposkytuje, krom snímků ve kterých je znázorněna trajektorie a střed objektů, jiný grafický výstup, nemělo smysl dále porovnávat

programy CTRAX a DROSANA mezi sebou, protože dále již můj program není schopen konkurence, oproti výše uvedenému programu v doposud uvolněné poslední verzi.

Při srovnání času potřebného k analýze se misky vah přiklání na stranu programu CTRAX. Test prokázal, že na testovacím PC se 5 minutové video s jedním objektem (mouchou) analyzovalo cca 50 min (30 minut trvala samotná analýza +20 minut trvala tvorba výstupního videa), kdežto programu DROSANA trvala analýza a tvorba výstupního videa 75 min. Avšak pokud by šlo o video s více objekty, vítězství by připadlo na stranu programu DROSANA, poněvadž čas potřebný k analýze snímku u programu CTRAX úměrně roste s počtem objektů ve scéně, v mém programu tento fakt nehraje žádnou roli a video je zpracováváno neustále stejnou rychlostí. Testovací PC bylo osazeno 1,6 Ghz procesorem Intel, 2 GB RAM operační paměti a operačním systémem Windows 7 (32 bit).

Díky velkým odlišnostem v účelech software proto uvádím stručné porovnání společných znaků detekčních schopností obou programů včetně jejich výhod a nevýhod v následujícím krátkém přehledu:

#### **CTRAX:**

##### Výhody:

- Detekce více parametrů pohybu (úhel otočení, velikost blobu)
- Možnost uživatelského výběru regionu
- Program šířen zdarma (včetně zdrojových kódů)
- Rychlá detekce, díky knihovnám NumPy a SciPy
- Program vyvíjen pro všechny dostupné platformy OS

##### Nevýhody:

- Nepřehledné grafické rozhraní, bez manuálu nenastavitelné!
- Pro zpracování výsledných dat je potřeba program MATLAB
- Data jsou v matici uložena odděleně
- Zvláště při analýze více objektů zmatečná detekce
- Ne příliš dobře navržené rozhraní pro opravu chybné detekce objektů
- Nároky na PC při tvorbě videa s vizualizací

#### **DROSANA:**

##### Výhody:

- Program šířen zdarma
- Nenáročný na HW
- Data jsou poskytována v textové podobě (CSV či volitelně XLS)
- Data jsou poskytována pohromadě
- Univerzální výběr detekovaného prostoru (misky, trubičky)
- Systém hlídání nekorektních dat
- Algoritmus detekce je dostatečně robustní proti chybám v detekci.

##### Nevýhody:

- Program dostupný pouze pro platformu Windows
- Program neposkytuje sofistikovaný grafický výstup dat
- Delší průměrná doba detekce na snímek oproti konkurenci

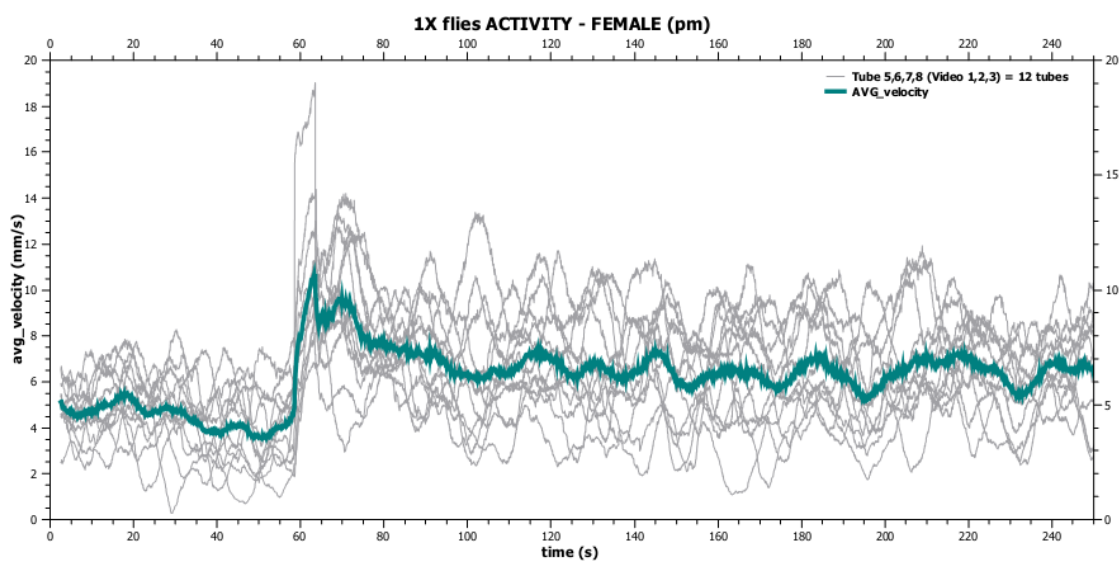
Dalším účelem programu DROSANA je měření pohybové aktivity, paměťových cyklů a reakčně-podnětových časů různých druhů mutací much, včetně doby uklidnění. Tento účel odlišný od všech dostupných účelů software, které jsou volně k dispozici a které popisují v kapitole 2, nebylo možné již s žádným dostupným software pro tento účel srovnat.

Jako vstupní zdroj dat bylo použito video s 8 trubičkami, v nichž se nachází předem stanovený počet much, v našem případě 10. Mouchy jsou předem uspany a přemístěny do trubiček, kde jsou po probuzení 15 minut nechány, dokud neodezní stres. V průběhu 5 minutového natáčení videa je po 60 s záznamu puštěn do jednotlivých trubiček nárazově proud vzduchu (2,5 atm), který trvá 200ms. U much je tak vybudena nadmíra stresu a vykazují nadměrnou pohybovou aktivitu hmyzu (v závislosti na typu much), program DROSANA měří míru vybudení (max. průměrnou rychlost v okamžiku externího podnětu), dále dobu za kterou se hmyz opět uklidní (dosáhne tzv. „baseline“, což je míra aktivity hmyzu, měřená v mm/s, která se vyskytuje především v bez stresových situacích). Opakováním měření je možné zjistit i vliv paměťového učení much na tento rušivý podnět (zda-li budou vykazovat snižující se tendenci pohybové aktivity). Program DROSANA umožňuje ve svém přednastavení určit, které snímky obsahují právě onen výše zmiňovaný, pro mouchy rušivý podnět. Indikace stavu aktivního proudu vzduchu je realizována pomocí LED diody umístěné vedle série trubiček, zobrazeno na obr. č. 4. Opakováním měření mezi různými typy much, je možné sledovat rozdílné behaviorální chování a účinky paměťového učení u různých typů much. Pro kontrolní test software byly použity 3 videa s 8 naplněnými trubičkami, z nich polovina byla vždy jiný druh much se samičím pohlavím (ke značení je používáno interních postupů BC AVČR, čili polovinu tvořily tzv. „control flies“ a druhá polovina „1X“). Výsledné hodnoty tedy jsou generovány z celkem 12 trubiček, v nichž bylo vždy 10 much.

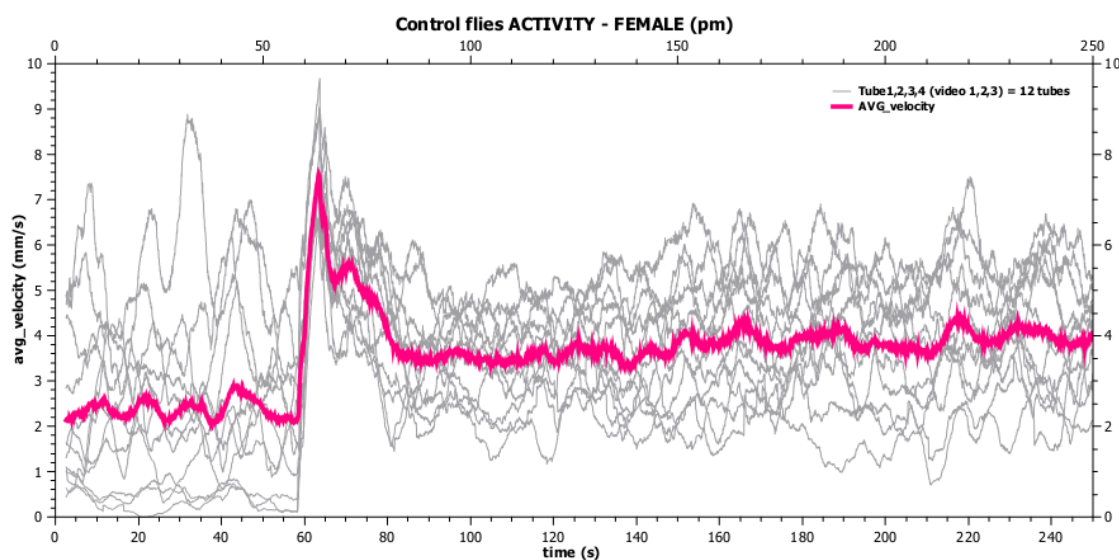


Obr. č. 4 Náhled pokusu při porovnání "1X" a "Control flies"

Jelikož bylo potřeba grafy aktivity much vyhladit, bylo použito tzv. kluzného průměru, což je v podstatě průměr z časových úseků (jejichž velikost je dána uživatelským nastavením, defaultně 5 s), které eliminují nežádoucí peaky v grafu. Data před vznikem rušivého podnětu a právě po něm jsou filtrována, tzn., že překročili-li hodnota kalkulované rychlost zadanou hranici (omezeno uživatelsky nastavitelným maximálním skokem, defaultně 25 mm/s), není tato rychlost do kluzného průměru započítána. V případě podnětu (kdy se všechny objekty v rámci jedno trubičky pohybují stejnou rychlostí danou proudem vzduchu) však je peak žádoucí, a data v tomto rozsahu jsou do kluzného průměru započítána. Výsledná data z měření jsou na následujících obrázcích:

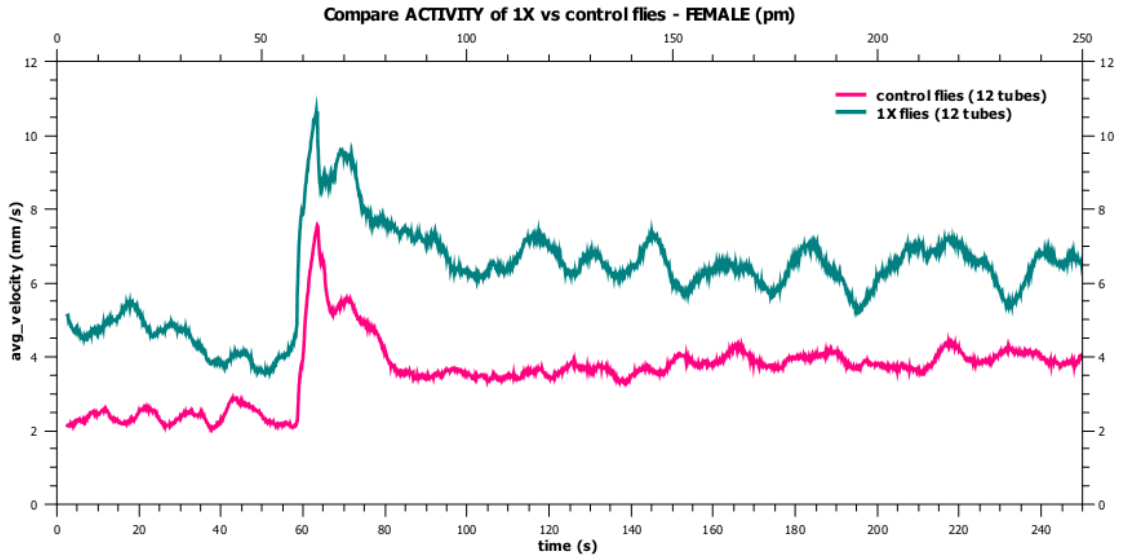


Obr. č. 5 Grafické znázornění pohybové aktivity 1X flies



Obr. č. 6 Grafické znázornění pohybové aktivity Control flies

Porovnáním křivek obou průběhů stagnující aktivity u výše zobrazených druhů much, je možné získat signifikantní rozdíl v behaviorálním chování těchto odlišných druhů much. Z obr. č. 7 je možné vyčíst, že křivky se liší nejen v klidové hodnotě průměrné aktivity, ale postupné uklidňování je realizováno odlišným způsobem s jinou hladinou pohybové aktivity.



Obr. č. 7 Porovnání křivek pohybových aktivit křivek 1X a Control flies

## 6 ZÁVĚR

Cílem mé diplomové práce bylo navrhnout a zrealizovat přístroj, jehož funkcí bude sledování a vyhodnocování pohybové aktivity hmyzu. Samotná realizace hardware byla vyřešena pomocí lehké montážní konstrukce s webovou kamerou, která je nyní i nadále v průběhu měření obohacována o další důležité a podpůrné externí prvky (teploměr, topení, vlhkoměr atd.). Software DROSANA speciálně vyvinutý k tomuto kusu hardware, jež dohromady s hardwarovou konstrukcí tvoří komplexně vyvíjený přístroj, je však ve fázi dlouhodobého testování a i přes svou relativní jednoduchost poskytuje sofistikované a signifikantní data, které dokáží s přehledem konkurovat získaným datům od již hotových softwarových řešení na trhu.

Počáteční směřování k víceúčelovosti vyvíjeného software poskytuje dostatečnou robustnost při pokrytí mnoha různorodých požadavků pracovníků Entomologického ústavu Biologického centra AV ČR v Českých Budějovicích na úpravu a schopnosti vyvíjeného software DROSANA. Dosavadní struktura programu umožňuje realizaci i budoucích projektů (testování paměti, monitoring spánkové aktivity, testování učících technik, monitoring chování) bez nutnosti velkého zásahu do programu, jež umožní a pomůže biologům pochopit neurologické a genetické procesy uvnitř stavby malého, na pohled nepodstatného organismu *Octomilky* obecné (*Drosophila Melangostar*), jehož stavebním kamenem organismu je podobně jako u člověka DNA s geny. Pochopením těchto principů, jak ukáže čas, získají pracovníci možnost přispět svými teoriemi ověřenými v praxi k léčbě moderních civilizačních či genetických nemocí.

Předběžné výsledky přístroje ukazují, že je možné korektně a velmi pohodlným způsobem ověřit stanovené premisy pouhým klikáním myši připojené k počítači. Moderní rozvíjející se počítačová doba a technologie tak usnadňují práci mnoha vědcům na světě a zkracují tak čas potřebný k získání toužených výsledků jejich usilovné práce.

Jsem hrdý na to, že jsem se mohl podílet na realizaci tohoto přístroje a věřím, že nejen jednoduchá konstrukce přístroje, ale především vyvinutý software přispěje svým dílem k novým objevům a realizacím nových experimentů se zásadními výsledky. V samotném vylepšování programu plánuji pracovat i nadále, plánované změny jako je překlad uživatelského rozhraní programu do několika mezinárodních jazyků, ukládání naměřených dat do databáze, která bude přístupná odkudkoli, přičemž uživatel bude mít možnost získat pouze ty data, která on sám potřebuje, dále pak vývoj vlastního engine pro grafický výstup naměřených dat a v neposlední řadě neustálé optimalizace rychlosti programu přispějí k tomu, že si program DROSANA získá velmi silnou uživatelskou základnu po celém světě.

Produkované výsledky mého přístroje či struktura a filozofie softwarového řešení včetně algoritmů se mohou stát velmi cennou devizou při studiu či následném užití v praxi mnoha studentům ať už biologických tak i technických oborů.

## 7 PŘÍLOHY

**Příloha1:** 1xCD obsahující software DROSANA ve verzi 1.6 včetně knihoven