



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA STROJNÍHO INŽENÝRSTVÍ**

FACULTY OF MECHANICAL ENGINEERING

**ÚSTAV AUTOMATIZACE A INFORMATIKY**

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

**NÁVRH UŽIVATELSKY PŘIZPŮSOBITELNÉHO  
AUTOMATIZOVANÉHO SYSTÉMU VIZUÁLNÍ KONTROLY  
KVALITY PRO MONTÁŽNÍ LINKY**

DESIGN OF CUSTOMIZABLE AUTOMATED VISUAL QUALITY CONTROL SYSTEM FOR ASSEMBLY LINES

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. Silvano Martini**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. Pavel Houška, Ph.D.**

**BRNO 2017**



# Zadání diplomové práce

Ústav:	Ústav automatizace a informatiky
Student:	<b>Bc. Silvano Martini</b>
Studijní program:	Strojní inženýrství
Studijní obor:	Aplikovaná informatika a řízení
Vedoucí práce:	<b>Ing. Pavel Houška, Ph.D.</b>
Akademický rok:	2016/17

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

## **Návrh uživatelsky přizpůsobitelného automatizovaného systému vizuální kontroly kvality pro montážní linky**

### **Stručná charakteristika problematiky úkolu:**

Stále vzrůstající tlak na zajištění co nejvyšší kvality produktů zvyšuje požadavky nejen na kontrolu funkčnosti výrobku, ale i na vizuální kontrolu, která musí odhalit vady výrobku, které vznikají při manipulaci či nedodržení výrobního/montážního postupu. Pro vizuální kontrolu jsou používány různé systémy počítačového vidění, které, ve většině případů, umožňují detekovat jenom jeden problém. Cílem této práce je navrhnout systém, který umožní provozovateli přidávat a konfigurovat detekci více problémů, tak aby nebyl závislý na dodavateli software výrobní linky.

### **Cíle diplomové práce:**

1. Seznamte se s problematikou kontroly kvality na montážních linkách;
2. Analyzujte obvyklé problémy zachycované kontrolou kvality a metody pro jejich detekci;
3. Pro jednotlivé problémy navrhnete způsob automatizace a definujte parametry pro posouzení kvality;
4. Navrhnete rozhraní, které umožní uživateli volbu a konfiguraci detekovaných problémů;
5. Navrhnete a realizujete systém počítačového vidění pro vizuální kontrolu kvality.

### **Seznam literatury:**

DEMANT, Christian, Bernd STREICHER-ABEL a Peter WASZKEWITZ. Industrial image processing: visual quality control in manufacturing. Berlin: Springer-Verlag, 1999, 353 s. : il. 1 CD-ROM. ISBN 3-540-66410-6.

DAVIES, E. R. (E. Roy). Computer and machine vision: theory, algorithms, practicalities. 4th ed. Amsterdam: Academic Press, 2012, xxxvi, 871 s. : il. ISBN 978-0-12-386908-1.

ŠONKA, Milan, Václav HLAVÁČ a Roger BOYLE. Image Processing, analysis, and machine vision. 4th ed. Stamford: Cengage Learning, 2015, xxxv, 870 listů : ilustrace. ISBN 978-1-133-59360-7.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2016/17

V Brně, dne

L. S.

---

doc. Ing. Radomil Matoušek, Ph.D.  
ředitel ústavu

---

doc. Ing. Jaroslav Katolický, Ph.D.  
děkan fakulty

## **ABSTRAKT**

Diplomová práce se zabývá problematikou strojového vidění. Rešerše je věnována popisu hardwarových a softwarových komponent systému strojového vidění. V části praktické je popsán návrh softwaru, který umožňuje provádění několika úloh strojového vidění současně.

## **ABSTRACT**

The diploma thesis deals with the problematics of machine vision. The research is devoted to description of the machine vision system hardware and software components. In practical part is described design and realization of the software, which allows several tasks of machine vision to be performed at the same time.

## **KLÍČOVÁ SLOVA**

Strojové vidění, kontrola výrobků, kontrola kvality, zpracování obrazu, LabVIEW.

## **KEYWORDS**

Machine vision, product inspection, quality control, image processing, LabVIEW.



## **BIBLIOGRAFICKÁ CITACE**

MARTINI, S. *Návrh uživatelsky přizpůsobitelného automatizovaného systému vizuální kontroly kvality pro montážní linky*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2017. 71 s. Vedoucí diplomové práce Ing. Pavel Houška, Ph.D.





## **PODĚKOVÁNÍ**

Tímto bych chtěl poděkovat své rodině za velkou podporu poskytovanou po dobu vypracování této diplomové práce a vedoucímu, panu Ing. Pavlu Houškovi, Ph.D. za poskytnuté rady a konzultace.



## **ČESTNÉ PROHLÁŠENÍ**

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením Ing. Pavla Houšky, Ph.D. a s použitím literatury uvedené v seznamu literatury.

V Brně dne 26. 5. 2017

.....

Silvano Martini



# OBSAH

<b>1</b>	<b>ÚVOD.....</b>	<b>15</b>
<b>2</b>	<b>STROJOVÉ VIDĚNÍ .....</b>	<b>17</b>
2.1	Osvětlení.....	18
2.2	Objektiv .....	20
2.3	Kamera.....	22
2.4	Rozhraní.....	24
2.5	HW platforma .....	26
2.6	Reprezentace obrazu.....	27
2.7	Základní algoritmy předzpracování obrazu .....	28
2.8	Základní algoritmy zpracování obrazu .....	32
<b>3</b>	<b>BĚŽNĚ KONTROLOVANÉ PROBLÉMY .....</b>	<b>37</b>
3.1	Čtení 1D a 2D kódů .....	37
3.2	OCR.....	38
3.3	Měření rozměrů .....	39
3.4	Povrchové vady materiálů .....	40
3.5	Detekce objektů .....	41
<b>4</b>	<b>NÁVRH SYSTÉMU STROJOVÉHO VIDĚNÍ .....</b>	<b>43</b>
4.1	Specifikace požadavků na systém strojového vidění .....	43
4.2	Návrh struktury systému strojového vidění.....	44
4.3	Analýza možností přizpůsobení systému strojového vidění.....	46
<b>5</b>	<b>REALIZACE SYSTÉMU STROJOVÉHO VIDĚNÍ.....</b>	<b>49</b>
5.1	Objektový model .....	50
5.2	Popis použitých tříd .....	51
5.3	Popis funkce systému .....	53
5.4	Uživatelské rozhraní .....	56
<b>6</b>	<b>UKÁZKA VYUŽITÍ SYSTÉMU .....</b>	<b>57</b>
<b>7</b>	<b>ZÁVĚR .....</b>	<b>65</b>
<b>8</b>	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>67</b>
<b>9</b>	<b>SEZNAM OBRÁZKŮ .....</b>	<b>69</b>
<b>10</b>	<b>SEZNAM PŘÍLOH.....</b>	<b>71</b>



# 1 ÚVOD

Kamerové systémy a digitální zpracování obrazu hrají v dnešní době čím dál podstatnější úlohu ve výrobním průmyslu, především pak ve velkosériové výrobě. Je tomu tak nejen kvůli automatizaci samotného výrobního procesu ale i automatizované kontrole kvality na výstupu. Tento trend je možné pozorovat nejen díky vzrůstajícím nárokům na kvalitu, ale také kvůli dokumentaci a normám, jako je například soubor norem ISO 9000. Tyto normy v mnoha případech bez automatizované kontroly kvality není možné dodržet. To následně může konkrétního výrobce značně znevýhodnit například ve výběrovém řízení na dodavatele konkrétních produktů.

Vizuální kontrola kvality výstupních výrobků byla dlouhou dobu prováděna školenými kontrolory, jelikož se však jedná o činnost silně rutinního charakteru, je tato činnost pro člověka náročná na udržení pozornosti. S tím je spjatá i náročnost na zamezení postupně se zvyšující náklonnosti k odbývání kontroly. Dalším problémem vizuální kontroly člověkem, který může v mnoha případech nastat, je nemožnost kontroly každého kusu samostatně. Tento případ může nastat díky automatizované výrobě, z důvodu rychlejší produkce oproti schopnosti člověka kontrolovat dané výrobky. V těchto případech se provádí kontrola pouze namátková, která však nemusí zachytit jednotlivé vadné kusy, ale pouze vady, které by se projevíly na celé sérii.

Pro spoustu výrobců může tedy být vhodné zvážit nahrazení člověka, v pozici kontrolora, zavedením systému automatizované kontroly kvality. Tato automatizovaná kontrola může přinést benefity v podobě zajištění kontroly celého objemu produkce kus po kusu, bezchybnost za správných provozních podmínek, či cenovou a výkonovou efektivnost. Pokud se však výrobce rozhodne pro zavedení takového systému, má před sebou úlohu výběru konkrétního automatizovaného systému optické kontroly, nebo jeho vlastní vývoj. Na trhu je v dnešní době nabídka jak úzce zaměřených stanic na specifické úkony (například kontrola plošných spojů), tak i modulární a uživatelsky konfigurovatelné systémy. Výhodou modulárních systémů je možnost relativně jednoduché modifikace stávající konfigurace, pro aplikaci na novou, třeba i velmi odlišnou úlohu. V ideálním případě s minimálními, nebo žádnými nároky na změnu hardwaru a bez nutné přítomnosti specializovaného pracovníka.

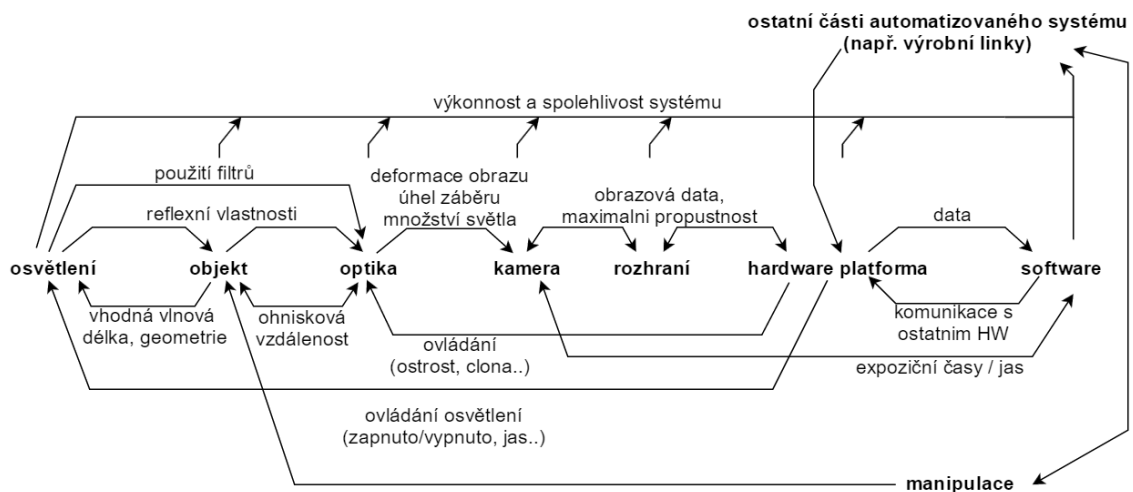
V případě zakoupení již hotového řešení systému strojového vidění je často nutné se spokojit s takovou podobou softwaru, která je nabízena, bez možnosti úprav. To však může být nežádoucí z důvodu speciálních požadavků na funkcionalitu, uživatelské rozhraní, či implementaci do rozsáhlejšího systému. V takovém případě je možné se buď dohodnout s vývojáři systému na jeho modifikaci, nebo si takovýto systém vyvinout nezávisle, čímž se zabývá právě tato diplomová práce.





## 2 STROJOVÉ VIDĚNÍ

Tato kapitola je věnována stručnému popisu základních stavebních kamenů systému strojového vidění. Strojové vidění, nebo též anglicky *machine vision*, je poměrně rozsáhlá disciplína, využívající poznatky z fyzikálního, strojního, softwarového, případně možno říci i matematického a elektro inženýrství. Zobecněně se pak dá říci, že se skládá z těchto částí: osvětlovače scény, objektivu, obrazového snímače, rozhraní pro přenos dat, výpočetního hardwaru, softwaru pro zpracování obrazových dat a případně ještě výstupního rozhraní pro předání dat získaných z obrazu pro další využití. Tyto části pak tvoří systém, ve kterém se vzájemně ovlivňují (Obr. 1) a při změně jedné části bývá nutné přizpůsobit i části ostatní. Při návrhu, či výběru jednotlivých částí je tedy nutné uvažovat vždy v kontextu celého systému. Jelikož se jedná o diplomovou práci zaměřenou na návrh softwarové části strojového vidění, je hardwarová část zmíněna v následujících kapitolách pouze stručně, avšak snad dostatečně aby bylo možné si z textu utvořit celistvý pohled na problematiku strojového vidění. V kapitolách věnovaných softwarové části pak jsou popsány některé základní algoritmy, z nichž bývají zkomponovány komplexnější funkce pro extrakci dat z obrazu.



Obr. 1 - Některé interakce komponent systému strojového vidění. (pouze výběr) [1]

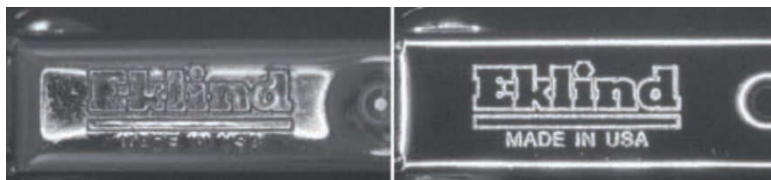
# HARDWARE STROJOVÉHO VIDĚNÍ

## 2.1 Osvětlení

Základem strojového vidění je světlo, protože právě ono je prvotním zdrojem obrazové informace, která je zachycena snímačem. Ačkoli na první pohled se může dodatečné osvětlení snímané scény zdát jako pouhý doplněk k okolnímu osvětlení, či podřadnou součástí systému strojového vidění, opak je pravdou. Dobře navržené osvětlení je základním předpokladem vysoké spolehlivosti systému a může i výrazně usnadnit práci softwarových inženýrů při návrhu algoritmů zpracovávajících obraz. Hlavním úkolem osvětlení totiž je zajištění co nejvyššího možného kontrastu ve snímané scéně a navíc, ideálně pouze na požadovaných prvcích snímané scény, přičemž nepodstatné prvky by neměly být kontrastní. Kontrast je totiž základem, na kterém jsou postaveny všechny algoritmy extrahující data z obrazu. [1] [2]

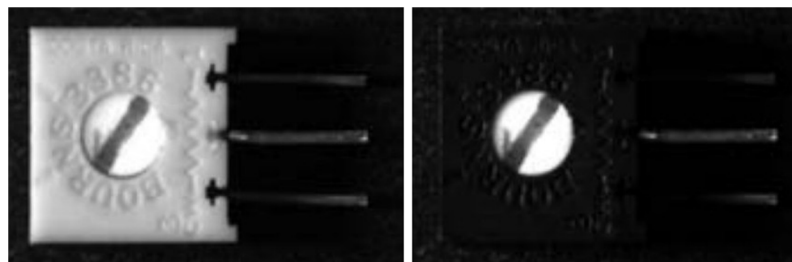
### 2.1.1 Přední aspekty návrhu osvětlení

- **Geometrie osvětlení** – Jednou z možností pro dosažení kontrastu mezi požadovanými prvky snímané scény je volba vhodné geometrie osvětlení, neboli v jaké pozici bude umístěno svítidlo vůči snímači obrazu a snímanému objektu. To má vliv na to, od kterých ploch snímaného objektu budou paprsky odraženy směrem ke snímači a které ne.
  - Přední osvětlení s jasným zorným polem (bright-field) – svítidlo je umístěno v malém úhlu vůči ose snímače, na snímač dopadá světlo odražené od kolmých, nebo téměř kolmých ploch.
  - Přední osvětlení s tmným zorným polem (dark-field) – svítidlo je umístěno ve velkém úhlu vůči ose snímače, na snímač dopadá světlo odražené od hran snímaného objektu, světlo dopadající na kolmé plochy je odraženo mimo snímač (Obr. 2).
  - Zadní osvětlení (backlight) – svítidlo je umístěno za snímaným objektem, světlo objektem buď prochází a tvoří kontrast pouze na jeho odlišně průsvitných prvcích, nebo je celý neprůsvitný a kontrast vytváří na jeho obrysech.



Obr. 2 - Porovnání různé geometrie osvětlení součásti s lesklým povrchem a vyraženým logem, vlevo bright-field, vpravo dark-field. [2]

- **Vlnová délka** – Vhodnou volbou vlnové délky světla, nebo též můžeme říci jeho barvou, je možné s výhodou využít u barevných součástí. Prakticky vzato, barva na snímané součásti shodná s barvou osvětlení se bude na snímku jevit světle, kdežto ostatní barvy se budou jevit tmavě a tak bude docíleno kontrastu (Obr. 3). Dále je také možné využít světla o vlnové délce mimo viditelné spektrum, tedy infračervené a ultrafialové, které v kombinaci s filtry před objektivem čerpají tu výhodu, že v běžném prostředí se nachází pouze málo zdrojů, které by osvětlení ovlivňovali. Poslední modifikací světelného záření, ačkoli nesouvisí úplně s vlnovou délkou, je polarizace světla a použití polarizačních filtrů před objektivem. Využitím polarizace je pak možno zachytit například vnitřní pnutí určitých průhledných materiálů.
- **Jas** – Jelikož na snímanou scénu může dopadat světlo i z okolních zdrojů světla, které jsou většinou nestálé, je nutné tyto rušivé vlivy eliminovat. Jako ideální eliminaci lze považovat úplné odstínění, to však není vždy technicky možné, v takové situaci je pak nutné zvolit osvětlovač s řádově vyšší intenzitou osvětlení [2]. Druhým důvodem pro volbu vysoké intenzity osvitu je požadavek na vysokou rychlost snímání, se kterou jsou spjaté krátké expoziční časy, přičemž požadované množství světla dopadající na senzor je stále stejné.
- **Homogenita** – Ačkoli existují metody pro detekci kontrastních přechodů, které jsou založeny na postupném porovnávání hodnot jasu, základní algoritmy, které jsou i rychlejší, pracují s absolutní hodnotou. Proto může být v některých případech nutností zajistit homogenní osvětlení. Například špatně použité bodové osvětlení by totiž mohlo způsobit nežádoucí výstupy algoritmů v oblasti okraje kuželu světla.
- **Tvar osvětlovače** – Pro dosažení různých geometrií osvitu při zachování homogenity osvětlení je možné na trhu zakoupit různé tvary osvětlovačů. Pro příklad vyjmenujme některé z nich: kruhová a pruhová v provedení dark-field i bright-field, matricová, kupolovitá a bodová.
- **Použitý zdroj světla** – Ve většině dnešních aplikací se používají jako zdroj světla LED diody díky dobrému poměru výkonu k ceně, odolnosti, reakčním časům, životnosti a dalším vlastnostem. Přesto však existují i jiné možnosti, které se mohou stále využívat kvůli některým svým charakteristickým vlastnostem, které LED diody nenahradí. Těmito alternativními zdroji jsou například halogenové, xenonové, nebo fluorescenční lampy, rtuťové výbojky, či laserové diody [1].



Obr. 3 - Modrý potenciometr osvětlen modře (vlevo) a žlutě (vpravo), snímán černobílou kamerou. [1]

## 2.2 Objektiv

Čočka, nebo soustava čoček, soustřeďující světlo na senzor se nazývá objektiv. V praxi se většinou již nevyužívají objektivы složené z jedné čočky, ale pouze objektivы složené z několika čoček, případně doplněné o clonu pro regulaci množství procházejícího světla. Pokud je objektiv vybaven i mechanismem pro posun čoček, tedy změnu ohniskové vzdálenosti, je tento objektiv označován jako schopný přiblížení neboli zoomu [3].

Pro účely strojového vidění je vhodné využití speciálních průmyslových kamer a objektivů, nikoli například webkamer, především z důvodu kvality vestavěné optiky. V mnoha případech je možno pozorovat označování objektivů pro průmyslové aplikace jako CCTV, v tom se však již konvenční značení rozchází s některou literaturou, kde se zkratkou CCTV označují pouze kamerové systémy s integrovaným objektivem [1].

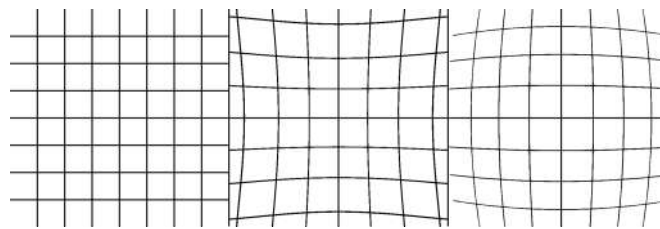
### 2.2.1 Parametry objektivů

- **Ohnisková vzdálenost** – jedná se o vzdálenost středu čočky od jejího ohniska. Touto vzdáleností je pak určen zorný úhel objektivu. Tento parametr ovlivňuje vzdálenost objektivu od snímané scény v závislosti na velikosti její plochy. Objektivы dělíme do skupin na objektivы s ohniskovou vzdáleností pevnou a variabilní. Pro účely strojového vidění se využívají převážně ty s pevnou ohniskovou vzdáleností.
- **Formát** – snímače kamer, umístěné za objektivem, se vyrábějí v různých velikostech a je nutné tuto velikost brát v potaz při volbě objektivu. Tento údaj bývá uveden v palcích a zjednodušeně řečeno udává, na jak velký snímač dokáže objektiv promítnout zobrazení. Volba objektivu většího formátu, než je velikost snímače, není problém, avšak v opačném případě nedokáže objektiv pokrýt celou plochu snímače a okraje výsledného obrazu budou černé. [4]
- **Clona** – jedná se o součást, která reguluje množství procházejícího světla objektivem a v podstatě jde o otvor s nastavitelným průměrem. Clona v kombinaci s expoziční dobou snímače pak nejen určuje světlost výsledného obrazu, ale také hloubku ostrosti, neboli vzdálenost od roviny zaostření, ve které jsou snímané objekty ještě ostré. Velikost clony se udává pomocí clonového čísla a značí se například  $f/2,8$ . Čím větší je clonové číslo, tím menší je otvor clony a méně světla clonou projde. [4]
- **Upevnění** – standardně se využívá dvou typů upevnění objektivů ke kamerám, tzv. C-mount a CS-mount. Obě varianty jsou vzájemně kompatibilní, jelikož mají stejnou velikost i jemnost závitu. Rozdíl je pouze ve vzdálenosti mezi snímačem kamery a přírubou objektivu. Pakliže se provede kombinace rozdílných typů uchycení na straně kamery a objektivu dochází ke změně optických vlastností oproti výrobcem udávaným, avšak to nemusí být závadou, pokud je tak činěno záměrně.
- **Speciální objektivы** – v aplikacích strojového vidění se též využívají speciální objektivы, jako jsou například telecentrické. Tyto objektivы na rozdíl od běžných nezachycují perspektivu, ale pouze ortogonální zobrazení. Využití telecentrických objektivů je především pro měření rozměrů objektů, kde by perspektivním zobrazením byla zanášena chyba vázaná na vzdálenost měřených bodů od objektivu.

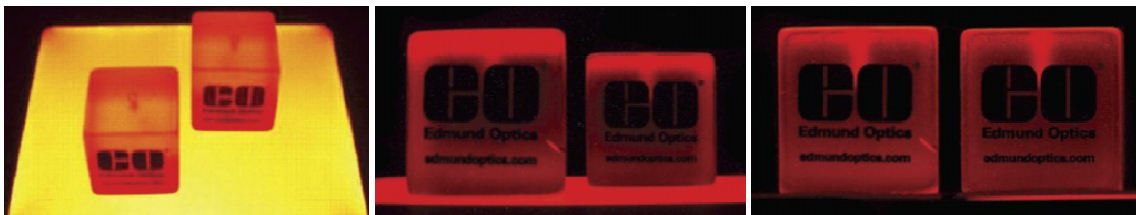
## 2.2.2 Optické vady

Existuje poměrně velké množství známých chyb vyskytujících se v obraze zapříčiněných objektivem. Na následujících řádcích však budou popsány pouze vady nejvýznamnější pro strojové vidění.

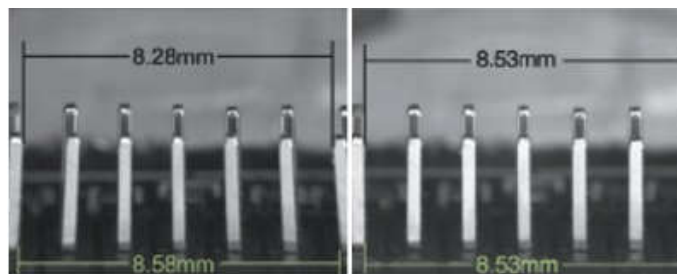
- **Perspektiva** – nejedná se až tak o vadu, spíše jako o nežádoucí jev. Vzniká zachycením trojrozměrné scény na dvojrozměrný snímač, především z krátké vzdálenosti. Vlivem perspektivy jsou vzdálenější objekty zachyceny jako menší než ty objektivu blízké. Pro zamezení perspektivního zobrazení je možné využít telecentrické objektivy [5] (Obr. 5), avšak zde je velmi limitující velikost snímané scény. Druhou možností je použití objektivu s velkou ohniskovou vzdáleností, umístěného ve větší vzdálenosti od snímaného objektu.
- **Geometrické zkreslení** – nejběžnější vady tohoto typu jsou soudkovitost a poduškovitost (Obr. 4). Tyto vady se projevují deformací obrazu, kdy u soudkovitosti je střed obrazu roztažen ke krajům a u poduškovitosti je naopak smrštěn do středu. Při výskytu kterékoli z těchto vad je možné provést softwarovou korekci, která se provádí pomocí kalibračního obrazce. Podoba kalibračního obrazce většinou bývá ekvidistantní, pravoúhlá mříž kontrastních bodů. Pokud je nezbytné, aby chyba ani nevznikla a nemusela se korigovat, je možné využít telecentrický objektiv (Obr. 6).



Obr. 4 - Geometrické zkreslení (originál, poduškovitost, soudkovitost)



Obr. 5 - Uspořádání scény, snímek běžným objektivem, snímek telecentrickým objektivem [5]



Obr. 6 - Použití telecentrického objektivu pro měřicí úlohy (vlevo klasický, vpravo telec. objektiv) [5]

## 2.3 Kamera

Dělení kamer je možné v první řadě provést na analogové a digitální, avšak analogové se již v dnešní době nevyužívají pro nové aplikace, pouze jako náhrada ve starých systémech, kde by přechod na kamery digitální byl nákladný. Z tohoto důvodu již nebudou analogové kamery ani rozhraní pro jejich připojení v této práci zmíněny.

### 2.3.1 Rozdělení digitálních kamer

- **Základní** (též nazývané „hloupé“) – Jedná se o kamery umožňující pouze snímání obrazu, bez jakéhokoli vyhodnocení. Možnosti nastavení těchto kamer zahrnují pouze konfiguraci parametrů snímače, jako je například expoziční doba, nebo parametrů základních úprav obrazu, jako je například jas, saturace barev a podobné.
  - **Vysokorychlostní** – Kategorie základních kamer zaměřená na snímání obrazu ve frekvencích řádově tisíců snímků za sekundu. Použití těchto kamer s sebou nese vysoké nároky na přenosové rychlosti a výpočetní výkon. Zpracování obrazu většinou neprobíhá v reálném čase, ale záznam je uložen do paměti kamery a zpracován až následně.
- **Chytré** – Kamery umožňující vyhodnocení snímaného obrazu. Výstupem těchto kamer většinou nebývá pouze obraz, ale i dodatečná data extrahovaná z obrazu. Dle možností vyhodnocení a konfigurovatelnosti můžeme dále dělit na:
  - **Konfigurovatelné** – Nastavení parametrů již připravených funkcí, například detekce pohybu, ovládání přisvícení, přepnutí režimu kamery a podobné.
  - **Skriptovatelné** – Volba algoritmů aplikovaných na obraze, způsobů jeho vyhodnocení a předávaných dat, avšak pouze pomocí softwaru výrobce.
  - **Programovatelné** – Nejvyšší variabilita, umožňují spuštění vlastního funkčního kódu.

### 2.3.2 Základní parametry

- **Rozlišení** – Základním parametrem všech kamer je jejich rozlišení, neboli počet obrazových bodů, které je schopný snímač zaznamenat. Udává se v pixelech.
- **Velikost senzoru** – Jedná se o velikost plochy snímající obraz. Společně s rozlišením má vliv na velikost jednotlivých pixelů, která hraje velkou roli ve schopnosti zaznamenávat obraz při nepříznivých světelných podmínkách.
- **Bitová hloubka** – Počet bitů použitý k popisu hodnoty jednoho pixelu, viz. kap. 2.6.
- **Snímkovací frekvence** – Udává počet snímků, který je kamera schopna zaznamenat během jedné sekundy.



- **Technologie snímače**
  - **CCD** – Náboj vzniklý fotoelektrickým jevem na jednotlivých pixelech snímače je sváděn postupnou změnou napětí na elektrodách vedených napříč snímačem ke kraji snímače, kde je stejným způsobem sváděn coby posloupnost elektronových shluků do výstupního zesilovače. Výstup čipu je analogový a tudíž je nutné osadit kameru dalšími elektronickými součástmi. Díky zmíněnému způsobu svádění náboje jsou CCD snímače energeticky náročnější.
  - **CMOS** – Každý pixel snímače disponuje vlastním zesilovačem, čímž je částečně zamezeno tvorbě šumu, vznikajícího při svádění náboje pomocí mříže vodičů. Postupným spínáním tranzistorů se zesílený náboj svádí k výstupnímu zesilovači a AD převodníku. Technologie výroby tohoto typu snímače je velmi podobná většině integrovaných obvodů, proto jsou výrobní náklady nižší než u CCD. Výstupem ze snímače jsou digitální data, a tudíž je jeho implementace snazší, než je tomu u CCD snímačů.
- **Typ závěrky** – Závěrka je původně mechanická část fotoaparátu, sloužící pro zamezení dopadu světla na snímač. U kamer se v dnešní době využívá převážně elektronická závěrka.
  - **Globální závěrka** (global shutter) – Všechny pixely jsou exponovány ve stejný čas, po stejně dlouhou dobu.
  - **Rotující závěrka** (rolling shutter) – Postupná expozice obrazu po řádcích, kdy první řádky jsou exponovány dříve a poslední později. U snímání pohybujícího se obrazu dochází vlivem rotující závěrky ke zkreslení pohybujících se objektů.
- **Metoda vyčítání obrazu**
  - **Progressive scan** – Vyčtení vždy celého obrazu postupně, po jednotlivých řádcích.
  - **Interlaced scan** – Periodické vyčítání lichých a sudých řádků samostatně, čímž je docíleno zdvojnásobení snímkovací frekvence. Výsledný obraz dané periody je složen z aktuálně vyčtených a jednu periodu starých obrazových dat. Tento typ snímání není vhodný pro analýzy pohybujících se objektů.
- **Provedení snímače**
  - **2D** – Nejrozšířenější provedení snímače nazývané area scan, kde jsou elementy snímače uspořádány do 2D matice.
  - **1D** – Tato varianta snímače, nazývaná line scan, má elementy uspořádány typicky v jedné řadě. Využití mají tyto snímače například při aplikaci nad pohybující se scénou, třeba pásovým posuvníkem, kdy je spuštěno kontinuální snímání a z jednotlivých pruhů se průběžně skládá výsledný 2D obraz.

## 2.4 Rozhraní

Různé kamery disponují různými možnostmi připojení, z kterých vyplývá jejich vhodnost použití a limity. V této kapitole jsou uvedena používaná rozhraní s některými jejich vlastnostmi.

- **Camera Link** – Sériová linka vyvinuta pro účely strojového vidění. Používá se ve třech provedeních lišících se propustností a počtem kabelů - base (až 255MB/s), medium (až 510MB/s) a full (až 680MB/s). Přes relativně nízké náklady na kabeláž, aplikace využívající tuto sběrnici patří mezi nákladnější, jelikož kamery s touto sběrnicí bývají většinou orientované na vysoký výkon a také je nutné použití digitálního frame grabberu. Maximální délka kabeláže je uvedena 10m. Z hlediska standardizace, Camera Link definuje pouze kabeláž, konektory, formát signálu a API pro konfiguraci kamer, nikoli však komunikaci mezi kamerou a počítačem, proto je nutné použití konfiguračních souborů pro konkrétní kameru. Sběrnice umožňuje napájení kamery, které je známé pod zkratkou PoCL (Power over Camera Link). CL díky použití frame grabberu téměř nezatěžuje procesor. Sběrnice dokáže vyhovět vysokým nárokům na synchronizaci díky flexibilnímu formátu datové komunikace. [6]
- **USB** – Sběrnice je použitelná od verze 2.0 avšak využívána je v dnešní době především ve verzi 3.0, na které je postaven standard USB3 Vision určený pro zařízení snímající obraz. Propustnost sběrnice je až 400MB/s. Jelikož je USB 3.0 rozšířeno téměř na všech PC a podobných výpočetních platformách, není nutné použít frame grabber a dokáže napájet připojenou kameru, lze její využití považovat jako nízkonákladové. Délka kabelu bez opakovače je do 5m. Sběrnice umožňuje přenos obrazu téměř s nulovým zatížením procesoru. [6]
- **IEEE 1394** – Sběrnice známá také pod označením FireWire jejíž propustnost závisí na verzi – IEEE 1394a dosahuje až 50MB/s a IEEE 1394b až 100MB/s. Náklady spojené s touto sběrnicí jsou nízké, bez nutnosti frame grabberu. Maximální délka kabeláže je u verze *a* 5m a u verze *b* až 100m, avšak doprovodným jevem je snížení propustnosti s rostoucí délkou. Z hlediska standardizace, před několika lety byl vytvořen standard pro komunikaci s průmyslovými kamerami, který podporuje několik video režimů a externích spouští. Zatížení procesoru závisí na ovladačích. FireWire sběrnice podporuje přímý zápis do paměti (DMA), při jehož využití není přenosem dat zatížen procesor, avšak nepodařilo se zjistit, zda se této funkce v praxi využívá. [1] [6]
- **Ethernet** – Hojně využívaná sběrnice především kvůli dostupnosti kabeláže a síťových prvků. Umožňuje jak přímé připojení kamery, tak i připojení do lokální sítě. Teoretická propustnost Gigabit Ethernetu je 125MB/s avšak díky mnoha vlivům snižujícím tuto rychlost není možné s touto hodnotou počítat, v praxi dosahuje rychlost stabilně kolem 40MB/s. Maximální délka kabeláže dosahuje až 100m bez opakovače. Napájení kamer je možné pomocí technologie PoE (Power over



Ethernet), avšak většina síťových karet nedisponuje dostatečným napájecím výkonem a tak pro využití PoE je většinou nutné použít síťový přepínač disponující touto technologií. Vytížení procesoru je závislé na ovladačích síťových karet. Přesná synchronizace při připojení do lokální sítě Ethernet prakticky není možná kvůli nepředvídatelnému provozu na síti.

- **GigE Vision** – Standard určený pro průmyslové kamery využívající síť Ethernet. Standard se využívá pro definici podoby komunikace. [6]
- **IP kamery** – Velmi rozsáhlá skupina kamer, využívající pro komunikaci síť Ethernet. Na těchto kamerách je spuštěn síťový server, ke kterému je možné se připojit a získat z něj snímaný obraz. Nevýhodou IP kamer může být fakt, že implementace serveru a způsobu získání obrazu není nijak standardizován.

## 2.5 HW platforma

Platforma, na které funguje řídicí software strojového vidění, může mít více podob. Kromě nejrozšířenější platformy PC, nebo průmyslových PC, je možné využít též platformy PXI, PAC, jednodeskových počítačů a jiné. Podstata funkce všech zmíněných platform je podobná, proto jsou v této kapitole zmíněny pouze jejich odlišnosti a vhodnost použití.

- **PC** – Nejotevřenější a nejrozšířenější platforma, díky čemuž nabízí výhody, jako je velký výběr z možností HW rozšíření, existence spousty podpůrného softwaru a dobrý poměr ceny a výkonu. Nevýhodou je nestálost nabídky komponent v čase, tudíž se může stát, že při nefunkčnosti některé součásti nebude možné pro výměnu pořídit stejnou. Při připojení do sítě se může jako nevýhoda jevit též zranitelnost systému, jelikož většina virů a jiných útoků je mířena právě na tuto platformu. [1]
- **IPC** – Upravená verze PC pro využití v průmyslu, vyznačující se většinou vyšším výkonem, spolehlivostí a odolností.
- **PXI** – Platforma odvozená od PC. Jedná se o modulární systém založený na sběrnici PCI, zaměřený na vysoký výkon a především velké datové toky systémem.
- **PAC** – Relativně nová platforma programovatelných automatů kombinuje vlastnosti PC a PLC. Vlastnosti, jimiž se blíží k PC, jsou zejména možnosti programovatelnosti a výkon. Kompaktní rozměry, 24V napájení, zvýšené krytí IPxx a spolehlivost při nepřetržitém nasazení jsou pak vlastnosti převzaté z platformy PLC.
- **PLC** – Programovatelné logické automaty většinou nedisponují dostatečným výpočetním výkonem pro složité algoritmy strojového vidění. Jejich využití je však možné v kombinaci se smart kamerami, optickými senzory či jinými prvky, které poskytnou danému PLC již zpracovaná data.
- **Vestavěné počítače (embedded)** – Jednoúčelové počítače, vyskytující se například ve smart kamerách. Tyto menší vestavěné systémy většinou disponují i nižším výkonem, jelikož obsluhují pouze jednu kameru, existují však i vestavěné počítače výkonem se vyrovnávající PC, navíc třeba doplněné i o FPGA.

# SOFTWARE STROJOVÉHO VIDĚNÍ

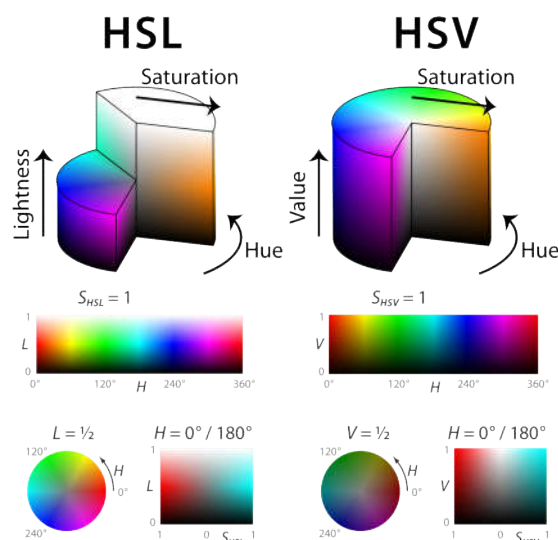
## 2.6 Reprezentace obrazu

Obraz zachycený kamerou bývá v počítači reprezentován v bitmapové podobě, nikoli vektorové. Bitmapový obraz je popsán jako 2D pole obrazových bodů, tzv. pixelů. Každý pixel nabývá určitých hodnot, které udávají jeho barvu dle použitého barevného modelu.

### Modely reprezentace barev – pouze výběr

- **Binární / monochromatické** – Pixely mohou nabývat pouze hodnot 0 a 1, většinou reprezentovány černou a bílou barvou. Reprezentace obrazu je správně nazývaná binární, a zobrazení jako monochromatické či černobílé.
- **Grayscale** – Stupnice šedi, pixel má pouze jednu hodnotu, udávající jeho světlost. Tento model bývá často mylně nazýván jako černobílý. V aplikacích strojového vidění se jedná o nejpoužívanější model.
- **RGB** – Barevný model složený z tří hodnot, udávajících intenzitu červené, zelené a modré barvy (red, green, blue). Výsledná barva pixelu je složena aditivním mícháním těchto tří barev.
- **HSV / HSL / HSI** – Barevné modely složené z tří hodnot. Ve všech případech první dvě písmena označují Hue (odstín), Saturation (sytost) a třetí písmeno znamená Value (hodnota), Lightness (světlost), nebo Intensity (intenzita) přičemž význam je prakticky stejný. Rozdílnost mezi modely spočívá v uspořádání jejich složek (Obr. 7). V počítačové grafice se využívá především pro svoji podobnost s lidským vnímáním barevného obrazu. [7]

**Bitová hloubka** – Někdy udávána jako barevná hloubka, je údaj, kolika bity je popsána barevná hodnota, z něhož přímo vyplývá, kolika různých odstínů může nabývat jeden pixel. Prakticky vzato, například 8 bitová hloubka umožňuje rozlišit 256 odstínů, 24 bitová hloubka pak 16.777.216 milionů odstínů.

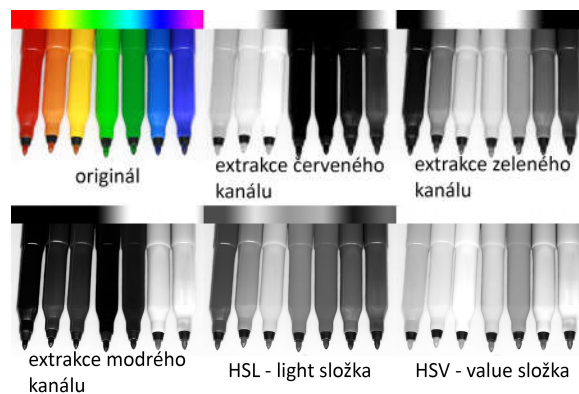


Obr. 7 - HSL a HSV barevné modely, autor: Jacob Rus

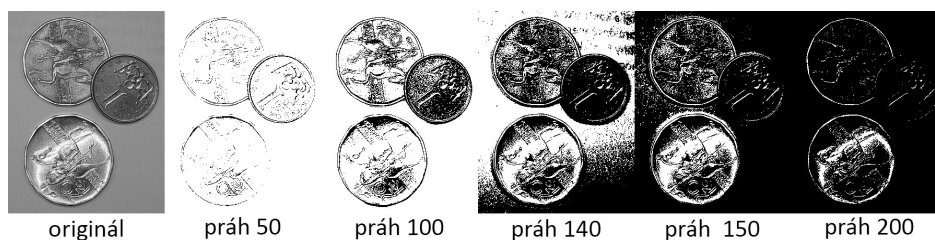
## 2.7 Základní algoritmy předzpracování obrazu

Před samotným zpracováním obrazu ve smyslu extrakce dat se mohou použít algoritmy pro přípravu obrazu do vhodnější podoby. Použitím správných algoritmů může být docíleno vyšší spolehlivosti, a ačkoli se jedná o operace navíc, tak i zrychlení celého výpočtu. Některé algoritmy zpracování by pak bez předzpracování nemohli ani fungovat. Pro úplnost je vhodné zmínit, že veškeré operace jsou matematicky založené, jelikož, jak již bylo zmíněno, jednotlivé pixely jsou pouhé číselné hodnoty. Jelikož existuje velké množství algoritmů a některé i v mnoha provedeních, na následujících řádcích budou uvedeny pouze některé z nich.

- **Převod barevného obrazu do stupnice šedi** – Možností převodu je několik, přičemž nejjednodušší je využití hodnot intenzity jednoho z kanálů RGB jako výsledných hodnot daného pixelu v odstínech šedi. Tento způsob však nebývá vždy vhodný a používá se jen v specifických případech. Častěji se využívá transformace do modelu HSL, HSV nebo HSI a extrakce hodnot se provádí z poslední složky, která popisuje světlost, čímž se výsledný obraz v odstínech šedi nejvíce podobá barevnému originálu. Ukázka je na Obr. 8.
- **Prahování** – Funkce pro změnu hodnoty pixelu dle nastaveného prahu. Většinou se provádí pouze binární prahování, kde pokud hodnota pixelu překročí hodnotu prahu, nastaví se na hodnotu 1 a pokud ne, nastaví se na hodnotu 0, nebo naopak. Pokud se provádí prahování z barevného obrazu, je nutné stanovit podmínky pro jednotlivé kanály samostatně, nebo stanovit matematickou operaci, jejíž výstupem bude opět jedna hodnota, která bude vyhodnocována. Ukázka prahování je na Obr. 9.



Obr. 8 - Převod barevného obrazu do stupnice šedi



Obr. 9 - Prahování

- **Vyhledávací tabulka** – Známejší pod anglickým názvem Lookup table (LUT), je tabulka obsahující pro vstupní hodnoty odpovídající hodnoty výstupní. Využívá se například pro rychlé vyhledání výstupní hodnoty určité funkce, jejíž spočtení je časově náročné. Kromě toho je také možné definovat vlastní hodnoty tabulky, které by funkcí byly jen těžko popsateľné. Ačkoli je možné použitím vyhledávací tabulky dosáhnout stejných výsledků jako u mnoha funkcí, má i podstatnou nevýhodu. Tato nevýhoda spočívá v nemožnosti adaptace hodnot pro konkrétní obraz, či lokální oblast, například při snížení světlosti, což předpis pomocí funkce umožňuje. [8]
- **Jas** – Zvýšení či snížení jasu je operace, kde se k původní hodnotě pixelu přičte, či odečte požadovaná hodnota, o kolik má být jas zvýšen, či snížen. Konkrétní algoritmus pak může přičítat či odečítat hodnotu nezávisle na poloze pixelu, nebo dle určité matematické funkce, například pro korekci vinětace. Při zvyšování jasu u RGB barevného modelu je nutno zvyšovat v určitém poměru všechny tři kanály. [9]
- **Kontrast** – Zvýšení či snížení kontrastu je v podstatě matematická operace, při níž se hodnoty ležící ve spodní polovině intervalu možných hodnot snižují a hodnoty ležící ve vrchní polovině intervalu se zvyšují. Hodnoty, které po této operaci leží mimo interval možných hodnot, se do tohoto intervalu zaokrouhlí. Ukázková funkce (1) je určena pro stupnici šedi s osmi-bitovou hloubkou, tedy 256 možnými odstíny.

$$\text{novýPixel}(\text{Faktor}) = \text{zaokrouhlení}((\text{Faktor} * (\text{původníPixel} - 128) + 128)) \quad (1)$$

Pro automatické zlepšení kontrastu je možno využít funkci (2), která funguje bez parametru a její princip spočívá v roztažení intervalu použitých hodnot pixelů daného obrazu na celý interval, který umožňuje bitová hloubka.

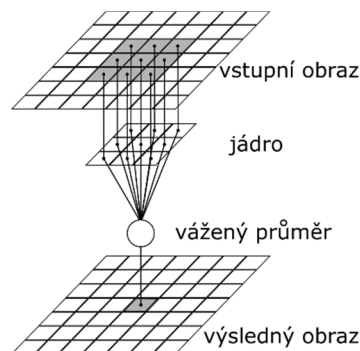
$$\text{novýPixel} = \frac{255}{\text{maxHodnota} - \text{minHodnota}} * (\text{původníPixel} - \text{minHodnota}) \quad (2)$$

Zmíněné funkce jsou ale pouze pro jednotnou úpravu kontrastu celého obrazu, což v mnoha případech nemusí být vhodné, či dostačující, zejména při nehomogenitě osvětlení. Proto se nezřídka využívá algoritmů pro lokální úpravu kontrastu, které vychází z rovnice (2), přičemž maximální a minimální použitá hodnota není brána z celého obrazu, ale pouze z určité lokální oblasti. Tímto algoritmem je pak možno odstranit například stíny, či plynulé přechody [8]. Algoritmů pro lokální úpravu kontrastu existuje více než pouze výše zmíněný, avšak jejich detailní rozbor není pro tuto práci podstatný.

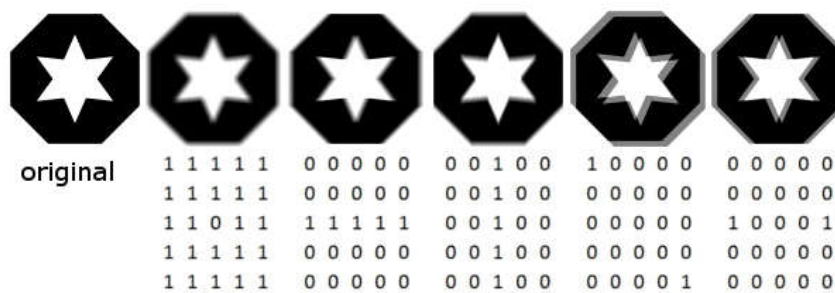


Obr. 10 - Ukázka zvýšení jasu a kontrastu

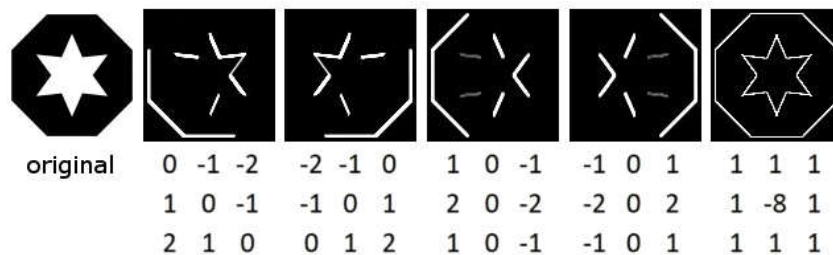
- **Obrazová aritmetika** – Ačkoli by se jako aritmetické operace dalo prohlásit mnoho funkcí, obrazovou aritmetikou jsou označovány funkce, kombinující dva či více obrazů. Jednotlivé aritmetické operace jako sčítání, odečítání, násobení a dělení jsou pak prováděny mezi jednotlivými pixely se stejnými souřadnicemi daných obrazů. Příklady použití jednotlivých operací jsou odečítání obrazů pro detekci rozdílů, násobení binárním obrazem pro maskování, maximum a minimum pro kombinaci obrazů s různým osvětlením, dělení pro vytvoření korekčního obrazu a jiné. [8]
- **Lineární filtry** – Jedná se o jedny z nejvyužívanějších operací, zahrnuté i v pokročilejších algoritmech, kde hodnota pixelu je určena v závislosti na okolních pixelech. Matice definující velikost okolí pixelu, jejíž velikost dimenzí nabývá lichých hodnot, se nazývá též jádro, nebo anglicky *kernel*. Hodnoty jednotlivých prvků jádra jsou váhami, pomocí kterých se následně vypočítá vážený průměr z hodnot pixelů jím odpovídajícím. Výsledek váženého průměru představuje novou hodnotu pixelu uprostřed tohoto jádra, viz Obr. 11. Pro aplikaci filtru na celý obraz je nutný výpočet váženého průměru pro každý pixel samostatně. Výsledný obraz je tedy závislý na hodnotách prvků jádra a jeho velikosti. Ukázky aplikace některých filtrů včetně použitých jader jsou na Obr. 12 a Obr. 13. [8] [9]



Obr. 11 - Princip lineárního filtru.



Obr. 12 - Ukázka lineárních filtrů pro rozmazání, včetně aplikovaných jader



Obr. 13 - Ukázka lineárních filtrů pro detekci hran, včetně aplikovaných jader



- Nelineární filtry** – Princip nelineárních filtrů je velmi podobný těm lineárním, avšak místo váženého průměru se využívá jiných matematických operací. Příkladem je možno uvést filtr založený na výpočtu mediánu, nebo skupinu filtrů nazývaných se morfologické. Morfologické filtry mají dvě základní podoby využívající buď operaci maximum, nebo operaci minimum. Složitější morfologické filtry jsou pak různými kombinacemi těchto dvou zmíněných. Jádro v případě morfologických filtrů má pouze binární hodnoty a definuje, které prvky budou do operace zahrnuty a které ne. Výstupem těchto operací je opět nová hodnota pixelu uprostřed jádra. V terminologii morfologie se jádro nazývá strukturální element. [8] [9]
- Sobelův, Robinsonův a Prewittův operátor** – Algoritmy aplikující lineární filtry pro detekci hran v různých směrech, čímž vzniká několik obrazů, z kterých se následně vytvoří, pomocí obrazové aritmetiky, obraz výsledný. Rozdíl mezi jednotlivými algoritmy je pouze v hodnotách prvků jádra a použité obrazové aritmetice. [9]
- Cannyho detekce hran** – Pokročilejší algoritmus, odolnější vůči detekci falešných hran způsobených šumem je složen z několika jednodušších algoritmů. Prvním krokem je aplikace lineárního filtru s jádrem o Gaussově rozložení pro rozmazání, neboli vyhlazení šumu. Krokem druhým je výpočet gradientů v obraze, který odhalí prudké přechody hodnot pixelů, tedy hrany. Následuje výpočet algoritmu „*non-maximal suppression*“, který by se dal popsat jako výpočet lokálních maxim ve směru kolmém k hraně, který převede plynulejší přechody na ostré hrany. Posledním krokem je nalezení vhodných hodnot pro dvojitě prahování do binární podoby a jeho provedení. [9]
- Frekvenčně založené filtry** – Jedná se o filtry typu horní, spodní a pásmová propust. Tyto filtry využívají konvoluce a zpětné Fourierovy transformace pro filtraci šumu v případě spodní propusti a detekci hran v případě horní propusti. [9]



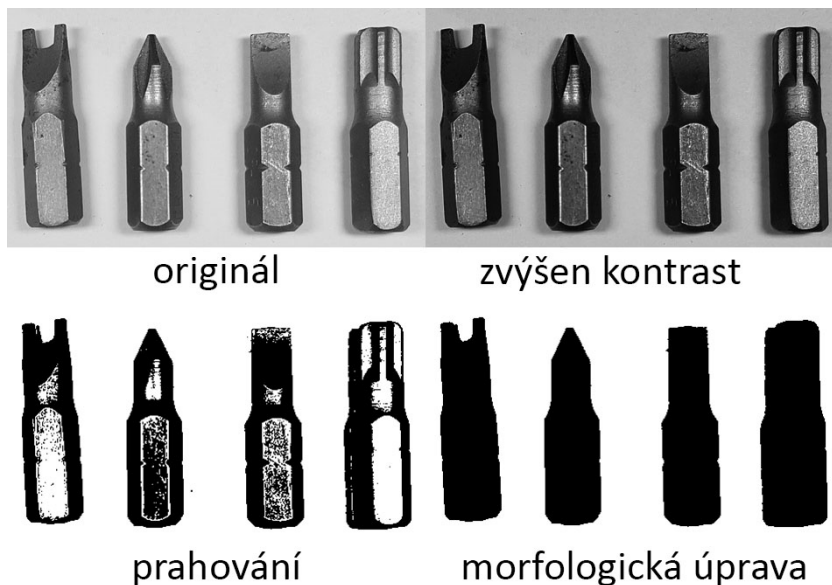
Obr. 14 - Sobelova a Cannyho detekce hran

## 2.8 Základní algoritmy zpracování obrazu

K analýze obrazu se převážně využívá algoritmů segmentace a klasifikace. Cílem segmentace je rozdělení obrazu na části reprezentující jednotlivé prvky snímané scény, cílem klasifikace pak kategorizace těchto prvků. V některých případech však není nutno použití jak segmentace, tak klasifikace, ale algoritmus může poskytnout výsledky bez druhého z páru.

### 2.8.1 Segmentace pomocí prahování

Výpočetně nejsnazší metoda, využívaná především pro separaci objektů od pozadí. Při využití prahování je vhodné zajistit kontrastní barvu pozadí vůči objektům a vysokou míru homogenity pozadí – například volbou vhodného osvětlení. Hlavní podmínkou pro úspěšnou segmentaci pomocí prahování je volba vhodného prahu. Nejspolehlivější metodou volby prahu je jeho manuální stanovení podle výstupu, avšak ne vždy je možné toto provést, zejména pokud není k dispozici zobrazovací zařízení, nebo by byla nezbytná jeho častá změna. V takových případech je vhodné použít adaptivní metody prahování, založené na analýze histogramu. Příkladem může být stanovení prahu pomocí procentuálního rozdělení zastoupení odstínů, využívané při prahování obrazu s textem, nalezením lokálního minima mezi maximy, nebo nalezením funkcí normálního rozdělení a výpočtem jejich průsečíku.



Obr. 15 - Segmentace pomocí prahování - výsledkem jsou čtyři segmenty



## 2.8.2 Segmentace na základě detekce hran

V této kategorii je zařazeno mnoho poměrně složitých algoritmů, jejichž základem jsou algoritmy pro detekci hran zmíněné v kapitole 2.7 - Základní algoritmy předzpracování obrazu. Cílem segmentačních algoritmů je vhodné seskupení detekovaných hran, které odpovídají reálným objektům, či prvkům v obraze. Pro docílení tohoto seskupení se nejdříve využívá filtračních algoritmů, které mají za cíl potlačit méně důležité, nebo falešné hrany (vzniklé například stíny objektu). Mezi nejznámější filtrační algoritmy patří například již zmíněný „*non-maximal suppression*“ převádějící hrany široké několik pixelů na hrany o šířce jednoho pixelu, filtrace hran podle jejich velikosti, nebo klasifikace a filtrace hran podle návaznosti na jiné hrany, nazývaná jako relaxace hran. Pro samotnou segmentaci se pak využívá opět několika možných algoritmů. Nejjednodušší algoritmy spočívají v nalezení oblastí ohraničených hranami a prohlášení výsledných seskupení za samostatné segmenty. Tyto algoritmy je vhodné použít, pouze pokud jsou objekty dostatečně odděleny. Při segmentaci obrazu obsahujícího mnoho navazujících hran se využívá algoritmů založených na převodu do grafové struktury a následné aplikaci algoritmů pro její prohledávání. Pro prohledávání grafové struktury se též využívá dynamického programování a Bellmanova principu optimality.

Posledním velmi významným způsobem segmentace obrazu dle hran je Houghova transformace, které se využívá pro vyhledávání předem definovaných geometrických tvarů v obraze. Houghova transformace umožňuje nalezení jak jednoduchých tvarů jako jsou přímky a kružnice (Obr. 16), tak i složitější křivky popsané polynomy. [9]



Obr. 16 - Houghova transformace - detekce kružnic

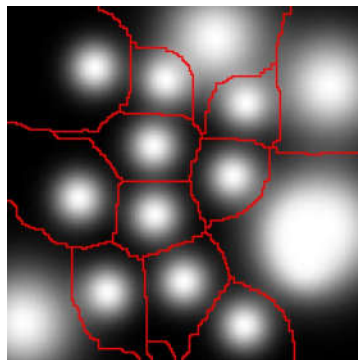
### 2.8.3 Segmentace na základě vlastností oblastí

Na rozdíl od předchozích metod, které hledali hranice mezi segmenty, následující metody vytváří segmenty přímou cestou. Hlavní myšlenkou těchto algoritmů je rozdělení obrazu na základě homogenity jednotlivých segmentů.

Základním algoritmem v této kategorii je růst a spojování segmentů, kdy v prvním kroku představuje každý pixel jeden segment. Postupným spojováním sousedních segmentů splňujících vhodné podmínky (týkající se homogenity) segmenty rostou až do chvíle, kdy nabydou maximální možné velikost a není možné je dále spojit bez porušení podmínky spojování.

Opačným algoritmem je postupné rozdělování, kde je výchozím stavem jeden segment o velikosti obrazu. V následujících iteracích se vždy porovná, zda by po rozdělení segmentu vznikly segmenty s rozdílnými vlastnostmi a pokud ano, učiní se tak. Samotný algoritmus rozdělování však vede k velkému množství segmentů, proto se jej využívá v kombinaci s algoritmem spojování, který se provádí buď při každé iteraci, nebo až po ukončení rozdělování. Výhodou kombinace těchto algoritmů je větší efektivita v případě výskytu velkých homogenních ploch v obraze, na druhou stranu v případě jisté symetrie je algoritmus rozdělování nefunkční. Nutno podotknout že zmíněné algoritmy jsou pouhým základním principem a na základě použitých podmínek spojování a rozdělování je možno vytvořit spoustu velmi odlišně fungujících variant.

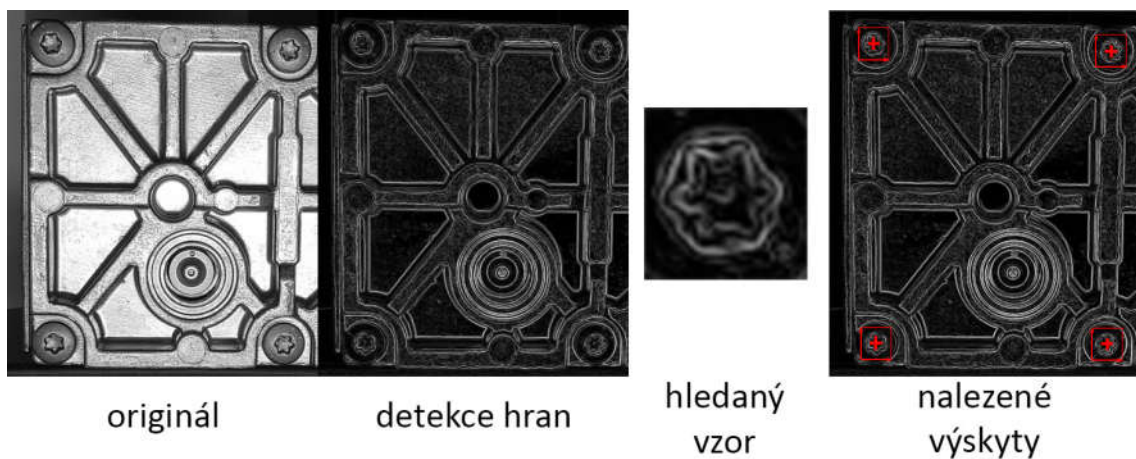
Posledním zmíněným algoritmem spadajícím do této kategorie je tzv. „*watershed*“ segmentace, která je převzata z oboru topografie (*watershed* = vodní předěl). Tento algoritmus je analogií k dělení vodních toků, či zádrži vody v závislosti na terénu. Existují dva základní principy výpočtu, prvním je určení příslušnosti pixelu k určitému segmentu, pomocí hledání lokálního minima, po cestě tvořené největšími gradienty. Všechny body končící ve stejném lokálním minimu jsou pak součástí jednoho segmentu. Tento způsob je však poměrně výpočetně náročný, proto se využívá spíše následujícího. Prvním krokem je nalezení lokálních minim, která představují počáteční stavy segmentů. Analogicky k stoupání spodních vod a zaplavování nejnižších míst se od nuly zvyšuje konstanta, dle které segmenty zabírají okolní pixely v každé iteraci. Pokud by mělo dojít ke spojení dvou segmentů, vytvoří se mezi nimi hranice. Obraz je rozsegmentován po dosažení hodnoty konstanty odpovídající nejvyšší hodnotě pixelu v obraze. Ukázka algoritmu *watershed* segmentace je na Obr. 17.



Obr. 17 - Watershed segmentace

### 2.8.4 Porovnávání vzorů / klasifikace

Skupina algoritmů s účelem vyhledání vzorů v obraze, či rozdělení detekovaných objektů do kategorií. Algoritmy jsou často založeny na porovnávání vlastností podoblastí obrazu, nebo segmentů vzniklých předchozími metodami, s vlastnostmi vzoru. Kritéria porovnávání pak mohou být od jednoduchých korelací až po složitější přístupy využívající grafových struktur. Pokud je vyhledávaným vzorem malý obraz, řádově o stovkách pixelů, není problém využít porovnávání matic pixelů přímo a hledat nejnižší odchylku. Při použití vzorů s větším počtem pixelu je vhodné provést převzorkování na nižší rozlišení. S tímto obrazem se provede hrubý odhad pozice a až následně se vyhledá vzor v dané oblasti, v plném rozlišení. V složitějších vzorech může být též výhodné najít výrazné prvky (hrany, rohy, díry), popsat jejich rozložení pomocí tzv. úplného grafu (existují hrany mezi všemi vrcholy) a následně hledat ve zpracovávaném obraze stejné, či do jisté míry transformované rozložení těchto prvků. Nutno podotknout, že porovnávání vzorů může být využito jak pro segmentaci, tak i pro klasifikaci již detekovaných segmentů z jiných algoritmů. Pro klasifikaci je možné využít klasifikátory využívající nejen vzory, ale například polynomy, či neuronové sítě. Ukázka algoritmu hledání vzoru je na Obr. 18.



Obr. 18 - Detekce vzoru - nalezení polohy šroubů



## 3 BĚŽNĚ KONTROLOVANÉ PROBLÉMY

### 3.1 Čtení 1D a 2D kódů

Pro značení výrobků se často využívá 1D a 2D kódů, protože umožňují uchování relativně velkého množství dat na malé ploše a především čtení pomocí čteček je rychlejší a spolehlivější oproti čtení klasického textu. Nejčastějšími zástupci těchto kódů jsou UPC a EAN v kategorii 1D kódů a Data Matrix, QR (Obr. 19) a PDF417 v kategorii 2D kódů.

Ačkoli se nejedná o algoritmy přímo kontroly kvality, či detekce problému, je zde čtení kódů uvedeno, jelikož se hojně využívá pro identifikaci a následně například pro účely statistiky, či dokumentace.

Velkou výhodou pro účely strojového vidění je skutečnost, že většina těchto kódů je tištěna černou barvou na bílé pozadí a disponují tzv. tichou zónou okolo kódu. Z této skutečnosti vyplývají předpoklady pro dosažení vysokého kontrastu při snímání a tím pádem i spolehlivosti.

Pro čtení kódu většinou není nutné zajistit speciální typy osvětlení, postačí pouze zajištění dostatečné intenzity, aby snímaný obraz nebyl příliš tmavý.

Algoritmus čtení kódu využívá prahování pro převod do binární podoby a následnou segmentaci na základní prvky kódu – čtverce a obdélníky v případě 2D kódů a čáry v případě 1D kódu. Tyto segmenty představují sekvenci binárních hodnot, které jsou následně dekódovány v závislosti na použitém kódu.

V případě snímání kódu pod určitým úhlem je vhodným předzpracováním obrazu geometrická transformace, pro korekci deformace kódu.



## 3.2 OCR

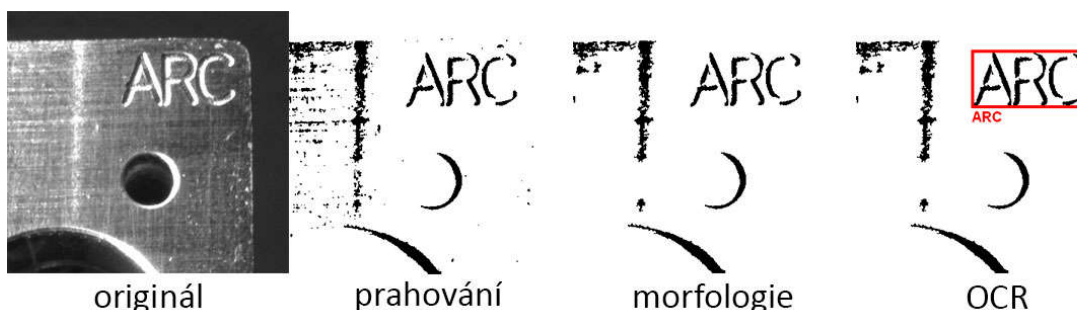
Dříve jediné a dnes stále běžně používané značení výrobků představuje text. Ačkoli je text pro strojové čtení textu náročnější než výše zmíněné kódy, nespornou výhodou je možnost jeho rychlého čtení člověkem i bez speciálních zařízení. Ve strojním průmyslu se pak hojně využívá vyraženého, či vyfrézovaného textového značení tam, kde by byl tištěný kód mechanickou manipulací, či tepelným zpracováním znehodnocen. Frézování 1D a 2D kódu se z důvodu náročnosti na přesnost často neprovádí. Algoritmus strojového čtení textu se přesněji nazývá optické rozpoznání znaků, anglicky *optical character recognition* – OCR.

Případy, kdy se OCR využívá pro čtení vyražených, či vyfrézovaných nápisů do lesklých povrchů, mívají poměrně vysoké nároky na osvětlení. Často je vhodné využít osvětlení typu dark-field z více směrů a tím „zviditelnit“ pouze hrany znaků. Nevýhodou osvětlení tohoto typu je vysoká citlivost na změnu úhlu osvitu.

Princip algoritmu spočívá v segmentaci obrazu na jednotlivé znaky a následné klasifikaci, neboli určení, o který znak ze znakové sady se jedná. Pro dosažení vysoké spolehlivosti je vhodné vytvořit znakovou sadu pro porovnávání z totožného fontu, jako je rozpoznávaný text. Občasným úskalím segmentace bývá přílišná těsnost znaků, čímž nedojde k rozdělení na jednotlivé znaky, ale pouze na skupiny znaků. V takovémto případě je možné buď zkusit nalézt vhodnější metodu segmentace, nebo do znakové sady přidat místo samostatného znaku jejich kombinaci, která bude odpovídat segmentu nerozdělených znaků. Algoritmus klasifikace bude tedy považovat určitý segment jako jeden například „dvoznak“.

Optické rozpoznávání znaků je poměrně výpočetně náročný algoritmus, proto v případech, kde je na snímané scéně větší množství textu, které není středem zájmu, ale pouze jeho určitá část, využívá se v praxi speciálních značek, od kterých je odvozena oblast, na kterou se omezí aplikace algoritmů. Tato oblast je označována jako ROI – anglicky *region of interest*, v překladu oblast zájmu.

Ukázka čtení textového označení součásti, vyfrézovaného do kovového povrchu je na Obr. 20.



Obr. 20 - OCR - kovový frézovaný povrch s frézovaným značením



### 3.3 Měření rozměrů

Úloha měření rozměrů se dříve prováděla výhradně mechanickými měřidly, či kalibry, později s nástupem automatizace se začali používat i optické senzory, avšak tyto technologie běžně umožňují měření pouze jednoho rozměru. Měření více rozměrů najednou je pak úloha, kterou je možné realizovat právě pomocí strojového vidění.

Jedná se o jednu z nejnáročnějších aplikací strojového vidění, jejíž náročnost spočívá ve vysokých požadavcích na přesnost a kalibraci. Pro dosažení vysoké přesnosti měření je nutné použití kamery s vysokým rozlišením, kvalitního objektivu s minimálním zkreslením a osvětlení zajišťující perfektní kontrast.

Výběr rozlišení kamery je odvozen od požadované přesnosti a velikosti snímané scény, respektive velikosti největších měřených rozměrů. Na první pohled by se mohlo zdát, že rozlišení by mělo odpovídat podílu velikosti snímané scény a tolerance přesnosti, avšak výsledek tohoto výpočtu je nutné minimálně zdvojnásobit, lépe ztrojnásobit, s ohledem na neostrost hran a měření rozměru v diagonálním směru, kde je hustota pixelů nižší.

Základním rozcestníkem při výběru objektivu by mělo být, zda se měření bude provádět mezi body ležícími v jedné rovině, či více rovinách. V případě jedné roviny, je možné použít libovolný objektiv s adekvátní kvalitou optiky. Pokud však je požadováno měření bodů ve více rovinách, výběr se značně zúží na objektivы telecentrické, které nezachycují perspektivu. Drobné geometrické vady objektivů je možné softwarově kalibrovat.

Většina aspektů volby osvětlení je zmíněna v kapitole 2.1.1, zde je vhodné zmínit především vhodnost osvětlení typu backlight, neboli podsvícení, v případech kdy se měří pouze celkové rozměry součásti.

Algoritmus pro měření rozměrů může mít mnoho podob, základem však je výpočet Euklidovské vzdálenosti mezi dvěma body. Tyto dva body je možné získat například těmito způsoby:

- detekcí hran na definované úsečce (měření rozměrů u přesně umístěné součásti)
- nalezením vzorů (měření rozteče šroubů, jako vzor použita hlava šroubu)
- nalezením středů kružnic pomocí Houghovy transformace (měření rozteče děr)
- nalezením ohraničujících úseček pomocí Houghovy tr. (měření rozměrů u součásti libovolně ležící na scéně)

Měřit se však nemusí pouze rozměry, ale třeba i plochy. V takových případech je možno využít vhodného způsobu segmentace a následně vypočítat plochu požadovaného segmentu.

### 3.4 Povrchové vady materiálů

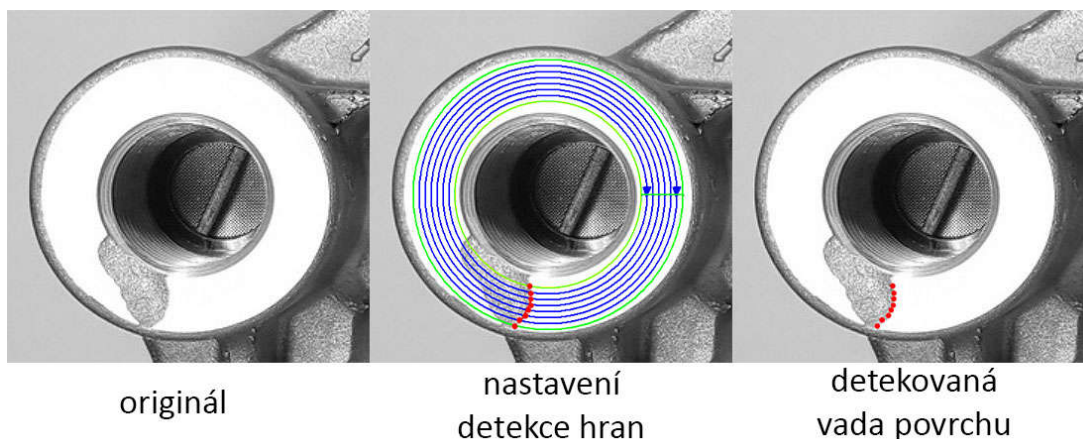
Kontrola povrchových vad se provádí u výrobků a materiálů jak z estetických, tak funkčních důvodů. Příkladem je možno uvést škrábance na designových doplňcích, skvrny na textiliích, či odtržení obráběných ploch po frézování (Obr. 21).

Povaha úlohy kontroly povrchových vad je opačného charakteru než všechny doposud zmíněné, jelikož součást je v pořádku, právě když není nic nalezeno. Z toho plynou jak výhody v podobě jednoduchosti algoritmů (pokud povrch nemá pravidelnou texturu), tak i nevýhody v podobě náročnosti na osvětlení a nastavení citlivosti, pro detekci případných vad.

Výběr kamery pro tyto úlohy by měl spočívat především ve volbě rozlišení, které závisí na jemnosti vad, které je potřeba detekovat. Kromě klasických 2D kamer je též možné s výhodou použít 1D kamery, které můžou kontrolovat například odvíjející roli materiálu. Většina aplikací nemá speciální nároky na kvalitu objektivu, ve smyslu geometrické přesnosti, pouze je nutná perfektní ostrost obrazu.

Osvětlení je možné využít všech typů, dle charakteru snímané scény. Příkladem je možno uvést vhodnost osvětlení typu dark-field pro detekci vad na hladkých površích, backlight pro průsvitné a průhledné materiály, případně i neprůsvitné v případě detekce protržení a bright-field pro detekci vad na členitějším povrchu.

Obecně je možné říci, že algoritmy jsou založeny na kontrole homogenity určité oblasti. Konkrétní algoritmus kontroly pak může spočívat například v detekci hran na určité ploše, vyhodnocení histogramu, či prahování. Pokud má však povrch pravidelnou texturu, je možné tyto algoritmy použít pouze pokud detekované vady budou kontrastnější než textura. V opačném případě je nezbytné použít pokročilejší algoritmy, které buď texturu odfiltrují, nebo dokáží detekovat odchylky. Takové algoritmy využívají například Fourierovu nebo wavelet (vlnkovou) transformaci.



Obr. 21 - Detekce povrchových vad pomocí detekce hran - održený kousek materiálu při frézování



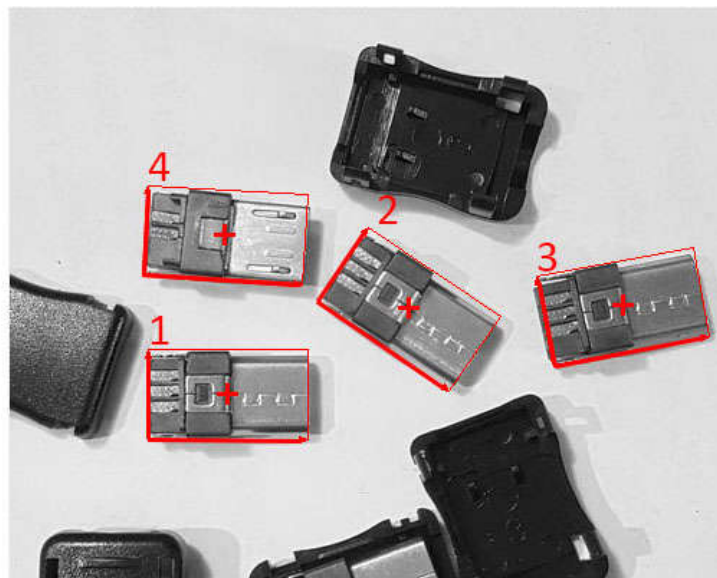
### 3.5 Detekce objektů

Kontrola přítomnosti a počtu objektů, nebo jejich pozice a natočení, to vše jsou varianty detekce nejen celých objektů, ale i jejich jednotlivých prvků. Tyto informace mohou dále sloužit jako referenční body pro další zpracování obrazu, nebo pro přímé vyhodnocení zda je snímáný objekt v pořádku. Příkladem je možno uvést kontrolu přítomnosti a počtu šroubů na určité součásti, natočení součásti pro kontrolu dotažení, třízení součástí, či detekci polohy pro navádění robotického manipulátoru.

Úlohy detekce objektů mohou být velmi různorodé, proto není na místě rozebírat vhodnost, či nevhodnost kamer, objektivů i osvětlení.

Algoritmy detekce objektů jsou běžně založeny na hledání vzorů, nebo segmentaci. V případě segmentace je výstupem plocha zabraná objektem na scéně a její pozice, většinou určená těžištěm nebo ohraničením plochy. Algoritmy založené na segmentaci je tak vhodné využívat pouze na detekci celých součástí, nikoliv však na detekci konkrétních prvků, ačkoli nemožné to není. Výhodou těchto algoritmů je jejich funkčnost i bez vzorů, tedy například pro třízení neznámých objektů pouze dle velikosti. Algoritmy založené na hledání vzorů umožňují kromě pozice i detekci natočení (Obr. 22), což v kombinaci se známými rozměry vzoru značně rozšiřuje oblast použitelnosti těchto algoritmů.

Na Obr. 22 má detekovaný objekt číslo 1 hodnocení 1000 a úhel 0° jelikož byl použit jako vzor. Objekt číslo 4 má nízké hodnocení, jelikož je konektor otočen a tudíž se vzoru podobá méně. Nízké hodnocení objektu 3 je způsobeno nehomogenním osvětlením.



Results ...	1	2	3	4
X Position	151,50000	277,19238	426,08771	149,37077
Y Position	269,50000	209,97574	207,85699	158,04364
Angle	0,000000	325,84671	10,24480	356,81866
Score	1000,00000	827,48267	668,46442	440,92252

Obr. 22- Detekce objektů - konektory



## 4 NÁVRH SYSTÉMU STROJOVÉHO VIDĚNÍ

Systém strojového vidění (systém SV) je navrhován jako systém, který bude integrován do řídicích systémů výrobních linek nebo systémů pro automatizované testování. Použití na výrobních linkách má zahrnovat především kontrolu kvality výrobku, představující identifikaci dílů, kontrolu povrchových vad, kontrolu přítomnosti klíčových dílů a přítomnost typového štítku. Použití pro automatizované testování zahrnuje např. kontrolu pracovního prostoru stroje, detekci začátku úniku provozních tekutin nebo pouhý obrazový záznam testu.

V mnoha případech se jedná o několik analýz na jednom regionu, nebo různé analýzy v několika regionech obrazu jedné nebo několika kamer. V tomto se navrhovaný systém SV zásadně liší od komerčně používaných SMART kamer, které většinou umí pouze jednu analýzu v jednom obraze. Další významnou změnou, kterou tento systém SV umožní, je konfigurace kamer a analýz z řídicího systému linky/testeru a začlenění ovládání do test plánu výrobku.

Systém SV bude vyvíjen v NI LabVIEW s využitím modulu NI Vision. Provozován bude na PC nebo na real-time kontroléru, kompatibilním s NI LabVIEW Real-Time.

### 4.1 Specifikace požadavků na systém strojového vidění

Systém SV by měl zahrnovat následující funkcionalitu:

- Správu kamer
  - detekce a volba kamer (připojení pomocí různých rozhraní)
  - konfigurace kamer (čtení a zápis atributů)
  - snímání a správa obrazu
  - ukládání a načítání konfigurace kamery
- Správu operací předzpracování obrazu
  - vytváření, konfigurace, provádění a rušení operací
  - kontrola správné návaznosti algoritmů
- Správu vision úloh
  - vytváření úloh
  - konfigurace úloh včetně výběru ROI
  - provádění úloh nad zvoleným ROI
  - vyhodnocení úloh a předání výsledků nadřazenému systému
  - rušení úloh
- Uživatelské rozhraní pro výše uvedené body
- Testovací rozhraní, včetně benchmarkování výkonu

## 4.2 Návrh struktury systému strojového vidění

### 4.2.1 Ovládání kamery

Funkce, kterou by měla zvládat tato část systému je komunikace s kamerou zajišťující jak přenos obrazu, tak i konfiguraci kamery. Jak již bylo zmíněno v kapitole 2.4, existuje více rozhraní pro připojení kamery. Cílem návrhu je možnost připojení kamer pomocí různých rozhraní, a proto bude návrh koncipován tak, aby bylo možné provádět rozšíření konektivity pomocí dědičnosti a implementace prototypové třídy.

Prvotním cílem bude zajištění konektivity kamer podporujících GenICam (Generic Interface for Cameras). GenICam je standard poskytující programové prostředí pro komunikaci a konfiguraci kamer, připojených přes kteroukoli sběrnici.

V dalším kroku by měla být zajištěna konektivita s IP kamerami, které však nemají standardizované prostředí.

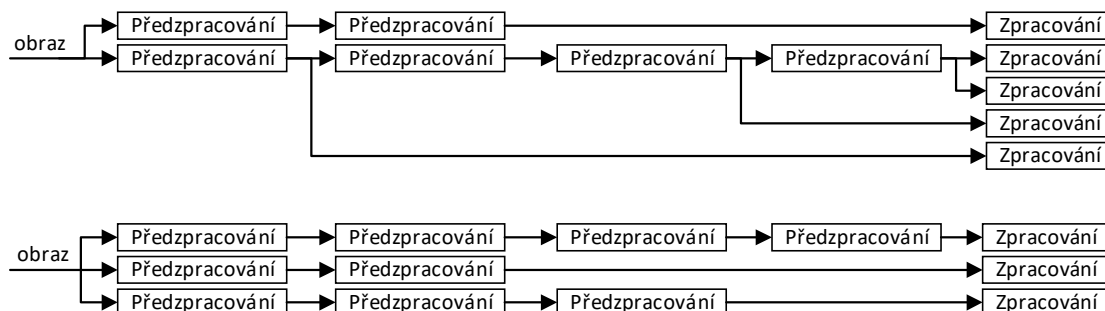
### 4.2.2 Zpracování obrazu

Jedná se o hlavní část systému strojového vidění, obhospodařující algoritmy pro předzpracování a zpracování obrazu, což ve své podstatě je zřetězení algoritmů za sebe.

Pro dosažení požadované funkcionality, tedy možnosti konfigurace sekvence těchto algoritmů je nezbytné jednotlivé algoritmy separovat do samostatných bloků. Při zřetězení těchto bloků je pak nezbytné kontrolovat jejich správnou posloupnost.

Koncepce struktury zřetězení může nabývat více podob, například několika paralelních sekvencí, nebo stromového rozvětvení jedné sekvence (Obr. 23). Větší pozornost konceptům zřetězení je věnována v kapitole 4.3.1. Návrh realizovaného systému bude veden směrem k první z variant, tedy samostatné sekvenci algoritmů předzpracování obrazu, zakončené jedním algoritmem zpracování.

Separace algoritmů do samostatných bloků bude řešeno pomocí prototypových tříd, samostatně pro algoritmy předzpracování a zpracování. Konkrétní algoritmy budou představovat děděné třídy těchto prototypů.



Obr. 23 - Různé zřetězení sekvence algoritmů

### 4.2.3 Vyhodnocování výsledků

Vstup a výstup jednotlivých operací předzpracování a úloh strojového vidění může mít různou podobu dle algoritmů, které úloha používá. Aby byla zajištěna jednoduchá rozšiřitelnost systému SV o nové algoritmy předzpracování a strojového vidění, musí být definována abstraktní třída VIOData (Vision Input/Output Data) včetně jejího rozhraní pro předávání dat a výsledků mezi jednotlivými operacemi a úlohami. Specifické třídy dat pro jednotlivé algoritmy budou odvozeny od VIOData.

### 4.2.4 Uživatelské rozhraní

Uživatelské rozhraní slouží k interakci uživatele se systémem. Touto interakcí, v případě navrhovaného systému, je konfigurace parametrů kamery, algoritmů a tvorba sekvencí algoritmů.

Návrh uživatelského rozhraní by měl mít za cíl dosažení co nejvyšší míry přehlednosti, při zachování všech možností konfigurovatelnosti. Hlavní roli hraje vhodná kategorizace konfigurovaných parametrů dle logické návaznosti. Při velkém množství ovládacích prvků je vhodné provést dělení do subkategorií. Tyto ovládací prvky pak mohou být umístěny v samostatných oknech, či panelech.

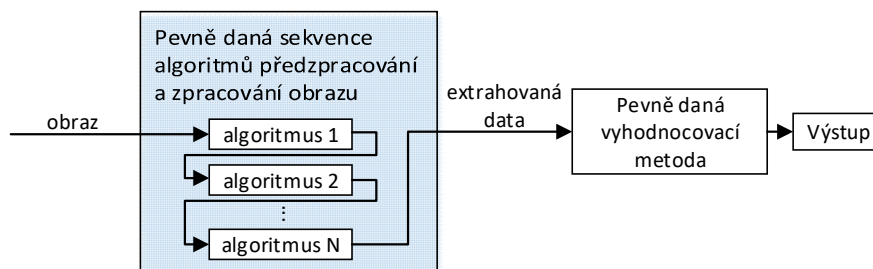
### 4.3 Analýza možností přizpůsobení systému strojového vidění

Před návrhem uživatelsky přizpůsobitelného systému SV jakéhokoli typu je nutné stanovit, do jaké míry má být přizpůsobitelný. Při zodpovídání této otázky je nutné brát v potaz, že se vzrůstající přizpůsobitelností systému, rostou nároky na kvalifikovanost obsluhy. Proto nemusí být vždy vhodné snažit se vytvořit systém, který bude možné detailně konfigurovat pro nesčetně úloh, když následně toho žádný z uživatelů nebude schopen využít. V takovém případě by daná míra přizpůsobitelnosti byla na obtíž a kontraproduktivní.

Druhým faktorem stojícím proti přehnané přizpůsobitelnosti je efektivita naprogramovaného kódu. Pro vytvoření přizpůsobitelného systému je nezbytné využívat konstrukce, jejichž režijní náklady prodlužují nezbytnou dobu od zachycení obrazu až po jeho vyhodnocení. Z tohoto hlediska je jednoduchý systém, který cyklicky provádí zachycení obrazu, sekvenci předzpracovávajících a zpracovávajících algoritmů rychlejší, než systém který v průběhu procesu musí kontrolovat uživatelské nastavení a dle něj volit, které části systému se do procesu zapojí a které ne.

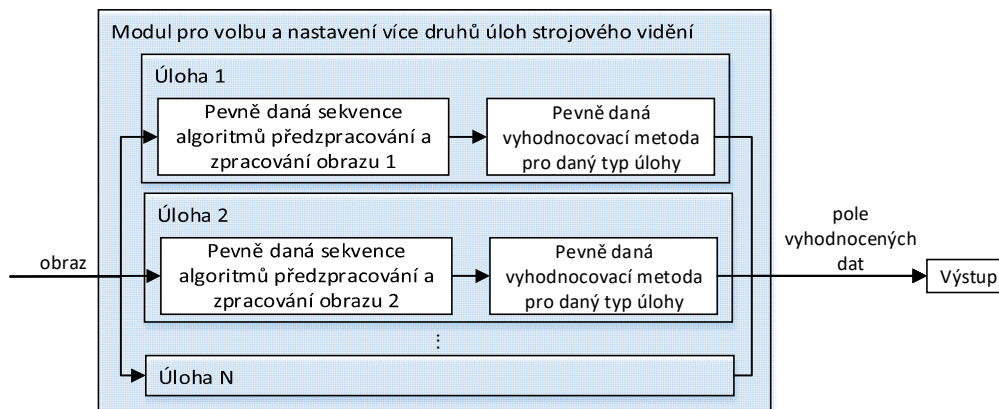
#### 4.3.1 Různé stupně přizpůsobitelnosti

Nejomezenější strukturou systému, umožňující pouze nastavení parametrů použitých algoritmů je pevná sekvence algoritmů (Obr. 24).



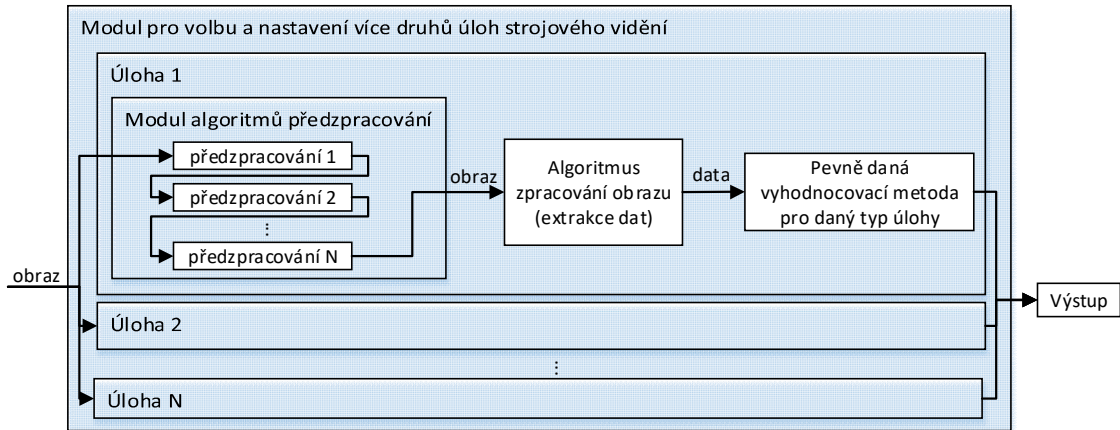
Obr. 24 - Systém strojového vidění – sekvence

Aplikace více pevných sekvencí (Obr. 25) je nejsnazší způsob jak docílit možnosti volby několika úloh, které mohou běžet současně. Za tímto účelem musí být vytvořen řídicí mechanismus.



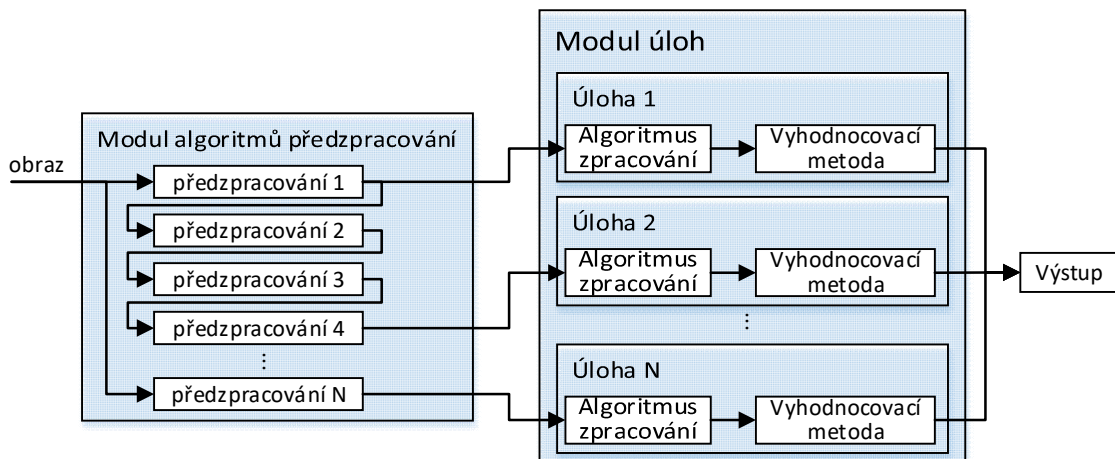
Obr. 25 - Systém SV - výběr několika úloh

Ve spoustě případů však může být nezbytná úprava sekvence předzpracování, kterou je možno poskytnout například tvorbou modulu předzpracování (Obr. 26). Takový modul by pak byl obsažen v každé z úloh.



Obr. 26 - Systém SV - několika úloh s modulem předzpracování

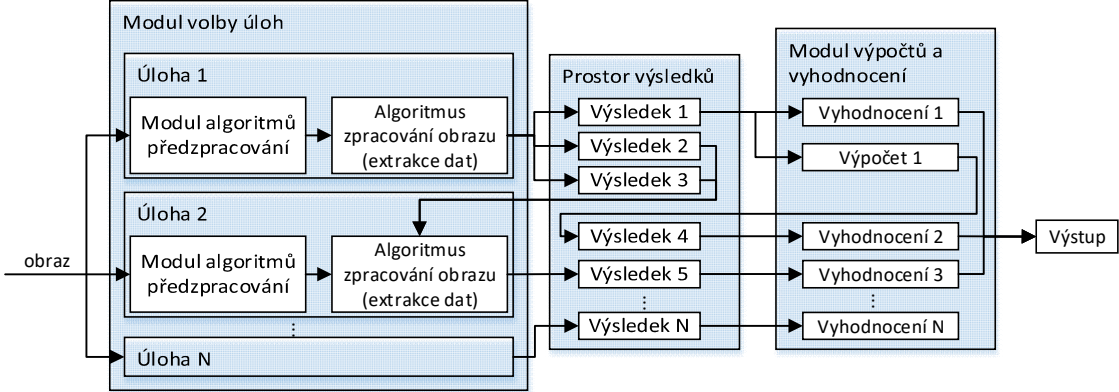
Alternativou k poslednímu zmíněnému může být konstrukce, kde existuje pouze jedna sekvence předzpracování, kterou je možno rozvětvit a která poskytuje mezivýsledky všech algoritmů předzpracování. Modul úloh by poté měl poskytnout možnost volby obrazu, který bude zpracováván.



Obr. 27 - Systém SV - jedna sekvence předzpracování, několik úloh

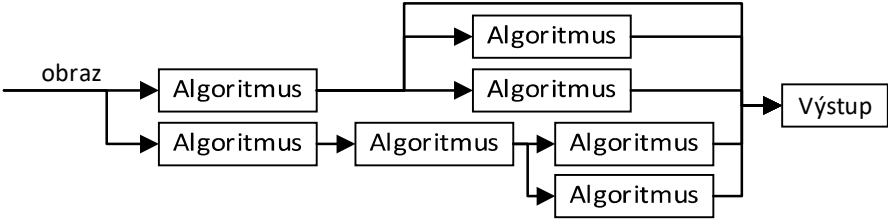


Dalším stupněm přizpůsobitelnosti je možnost zřetězení více algoritmů zpracování úloh za sebe, respektive využívat data extrahovaná jedním algoritmem pro definici algoritmu druhého. Toho je možné docílit například konstrukcí zobrazenou na Obr. 28, kde by existoval prostor, do kterého by všechny algoritmy zapisovaly své výsledky. Z tohoto prostoru by pak mohly čerpat data další úlohy, nebo modul pro výpočty a vyhodnocení. Největším úskalím tohoto konceptu však je zajištění správné posloupnosti vykonání jednotlivých algoritmů.



Obr. 28 - Systém SV s možností tvorby závislosti algoritmů a modulem výpočtů a vyhodnocení

Kombinací posledních dvou zmíněných a zároveň nejvyšším stupněm přizpůsobitelnosti je konstrukce zobrazena na Obr. 29. Jedná se o koncept, kde by byly všechny algoritmy děděny ze stejného prototypu a byly tudíž rovnocenné. To by za jistých předpokladů umožnilo jejich libovolné zřetězení (s kontrolou správné návaznosti). Tento koncept by tak nabízel možnost libovolného rozvětvení algoritmů předzpracování i zpracování a využití extrahovaných dat pro definici algoritmů následujících.



Obr. 29 - Systém SV - libovolné zřetězení algoritmů

Můj dlouhodobý cíl je dosáhnout nejvyššího stupně přizpůsobitelnosti, který je velmi komplexní a časově náročný na implementaci. Proto se z časových důvodů budu při realizaci této práce zabývat nižším stupněm přizpůsobitelnosti na úrovni několika úloh s modulem předzpracování.



## 5 REALIZACE SYSTÉMU STROJOVÉHO VIDĚNÍ

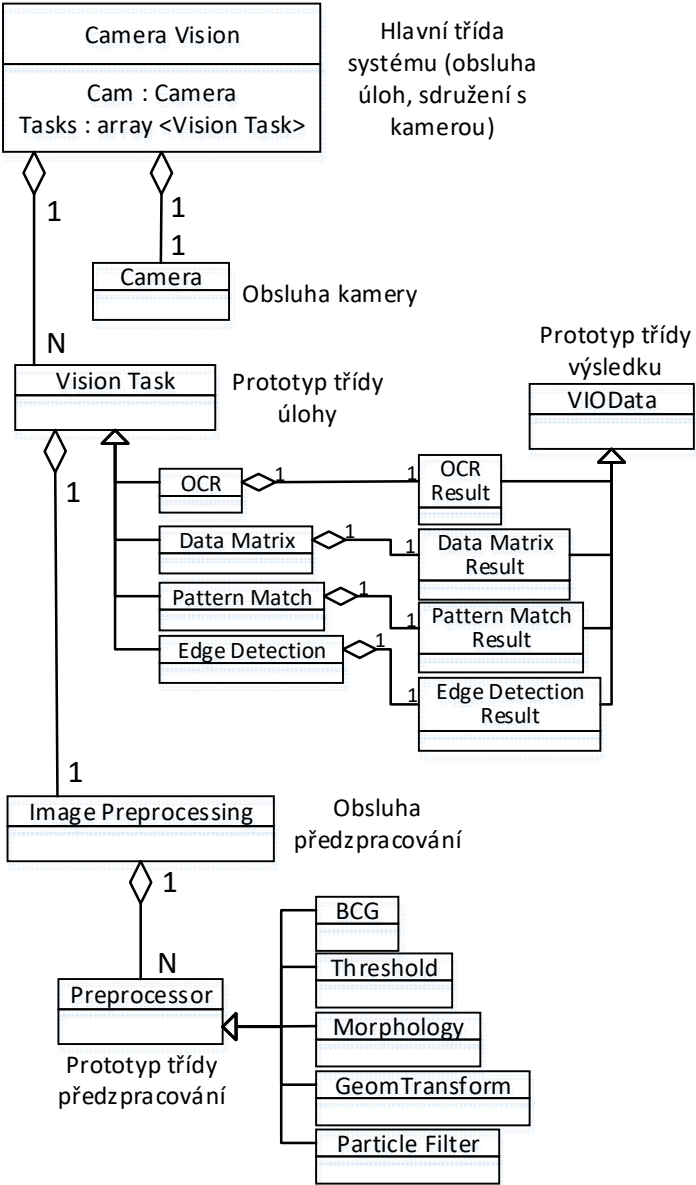
Realizovaný systém SV byl postupně vyvíjen od jednoduché sekvence, až do konečné podoby, s následujícími možnostmi:

- Konfigurace kamery
  - Detekce připojených kamer pomocí USB a GigE Vision
  - Možnost přepínání mezi kamerami za běhu systému
  - Vyčtení a zobrazení atributů kamery včetně jejich povolených hodnot
  - Možnost zápisu povolených atributů do kamery
- Konfigurace úloh strojového vidění
  - Nabídka čtyř úloh strojového vidění
  - Možnost navolení několika úloh běžících současně
  - Plně konfigurovatelná sekvence předzpracování pro každou úlohu samostatně
  - Ukládání konfigurace do xml souboru a jeho načítání
  - Pevně dané vyhodnocování výsledků pro každý typ úlohy, avšak s možností nastavení parametrů
- Konfigurace sekvence algoritmů předzpracování
  - Definována výchozí sekvence každé úlohy pro její správnou funkci
  - Možnost vkládání a odebírání úloh sekvence
  - Kontrola správnosti sekvence vzhledem k typu úlohy

Oproti navrhovanému konceptu systému, neumožňuje tento systém zřetězení více úloh strojového vidění za sebe, ale pouze zřetězení sekvence předzpracování, zakončenou jedinou úlohou zpracování.

### 5.1 Objektový model

Zjednodušený objektový model, bez uvedených atributů a metod jednotlivých tříd je znázorněn na Obr. 30. Výhodou tohoto návrhu je možnost jednoduchého rozšíření systému o nové úlohy strojového vidění, či algoritmy předzpracování obrazu pouhým vytvořením nové děděné třídy a implementací metod předka. V takovém případě není nutný téměř žádný zásah do hlavních řídicích mechanismů.



Obr. 30 - Zjednodušený objektový model systému

## 5.2 Popis použitých tříd

### 5.2.1 Camera Vision

Hlavní třída celého systému, jejíž objekt v podstatě obsahuje všechny použité objekty ostatních tříd a tudíž celou konfiguraci systému. Při běhu systému existuje pouze jedna instance objektu této třídy. Obsahuje metody pro řízení a konfiguraci systému na nejvyšší úrovni.

### 5.2.2 Camera

Účelem objektu této třídy je obsluha kamery, její volba a konfigurace. Metody určené pro obsluhu kamery jsou koncipovány tak, aby umožňovali rozdělení řídicího mechanismu do dvou částí. To znamená, že existuje skupina metod určená pro odesílání příkazů a druhá skupina pro zpracování těchto příkazů a obsluhu kamery. Metody pro odesílání příkazů jsou použity na mnoha místech systému, kdežto metody pro zpracování příkazů a obsluhu kamery běží v samostatné smyčce, tedy paralelně ke zbytku systému.

Výstupní obraz objektu této třídy je periodicky obnovován na stejném místě alokované paměti a poskytován zbytku systému pomocí reference.

### 5.2.3 Vision Task

Rodičovská třída ke všem úlohám strojového vidění. Děděné třídy představují konkrétní úlohy strojového vidění. Díky dědičnosti a virtuálním metodám je možné využívat tyto metody z řídicích částí systému, bez ohledu na konkrétní typ úlohy a vyvolávat konfigurační okna dle příslušných typů úloh.

Všechny úlohy si při inicializaci alokují vlastní paměť pro obraz, který zkopírují z výstupu třídy Camera a všechny operace provádí nad touto kopií. Tímto je zachován originální obraz pro využití ostatními objekty.

Virtuálními metodami určenými pro implementaci děděnými třídami jsou inicializace, konstruktor, destruktory, vykonání úlohy a konfigurační dialog.

#### Děděné třídy:

- OCR Task - Strojové rozpoznávání znaků. Kromě implementace metod předka je doplněna o metody pro vytváření znakové sady (učení) a její editaci.
- Data Matrix Task - čtení 2D kódu typu Data Matrix.
- Edge Detection Task - Detekce hran. Možno aplikovat na úsečce, obdélníkové a cirkulární ploše. Třída je rozšířena o metodu Results To Image pro vykreslení výsledků přímo do obrazu, pro komfortnější uživatelské nastavení.
- Pattern Match Task - Porovnávání vzorů. Třída je rozšířena o metody pro práci se vzory, tedy volbu z uložených vzorů, vytvoření vzoru ze snímaného obrazu a jeho učení v závislosti na zvolených parametrech.

### 5.2.4 Image Preprocessing

Třída fungující jako modul vykonávající předzpracování a všechny úkony s ním spjaté. Úkony prováděnými touto třídou jsou: tvorba a editace sekvence algoritmů předzpracování, vykonání této sekvence nad určeným obrazem a zajištění správnosti formátu výstupního obrazu. Objekt této třídy je součástí každé z úloh strojového vidění Vision Task.

### 5.2.5 Preprocessor

Rodičovská třída ke všem algoritmům využívaným třídou Image preprocessing. Děděné třídy představují konkrétní algoritmy předzpracování. Díky dědičnosti a virtuálním metodám je možné využívat tyto metody bez ohledu na konkrétní typ úlohy a vyvolávat konfigurační okna dle příslušných typů algoritmu.

#### Děděné třídy:

- BCG Preprocessor - Úprava jasu, kontrastu a gammy (BCG = brightness, contrast, gamma).
- Threshold Preprocessor - Prahování obrazu do binární reprezentace.
- GeomTransform Preprocessor - Geometrická transformace obrazu, využitelná např. pokud je obraz snímán pod úhlem a je potřeba jej „narovnat“.
- Morphology Preprocessor - Morfologická transformace obrazu, prováděná nad binárním obrazem.
- Particle Filter Preprocessor - Filtrování prvků z binárního obrazu.

### 5.2.6 VIOData

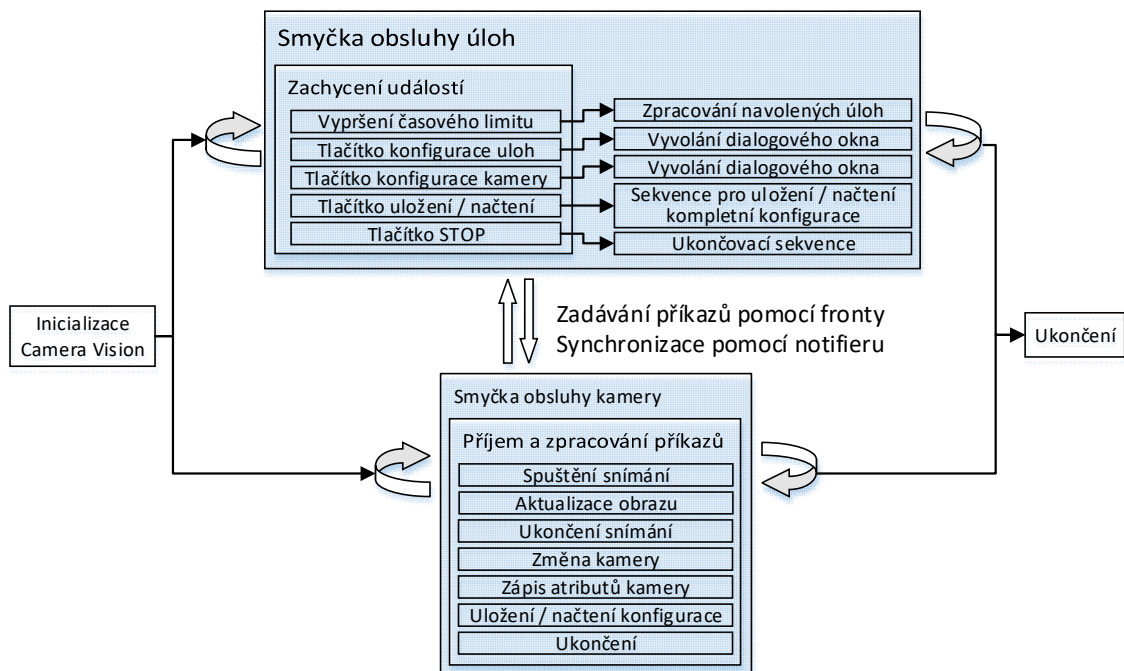
Rodičovská třída, jejíž děděné třídy nesou výsledky jednotlivých úloh. Každá z děděných tříd implementuje metody pro vyhodnocení výsledků a jejich výpis.

Implementace potomků této třídy byla provedena pouze pro výsledky potomků třídy Vision Task.

### 5.3 Popis funkce systému

#### Hlavní řídicí mechanismus

Hlavní řídicí mechanismus (Obr. 31) je rozdělen do dvou paralelních smyček, přičemž první je určena pro zpracování úloh a jejich konfiguraci a druhá pro snímání obrazu. Při inicializaci se vytváří objekt Camera Vision, přičemž se mimo jiné i alokuje paměť pro snímání obrazu, frontu a tzv. notifier, což je nástroj LabVIEW pro přenos zpráv mezi nezávislými částmi kódu, za účelem synchronizace. Obě smyčky tím pádem disponují referencemi jak na umístění obrazu, tak i na „komunikační kanály“. Účel tohoto rozdělení do dvou paralelně běžících částí je neustálá obnova obrazu v alokované paměti, nezávislá na právě spuštěné funkci smyčky obsluhující úlohy zpracování obrazu. Pokud by tak nebylo učiněno, musela by každá metoda, vyžadující obnovu obrazu, ovládat kameru pomocí příslušných metod sama.



Obr. 31 - Hlavní řídicí mechanismus

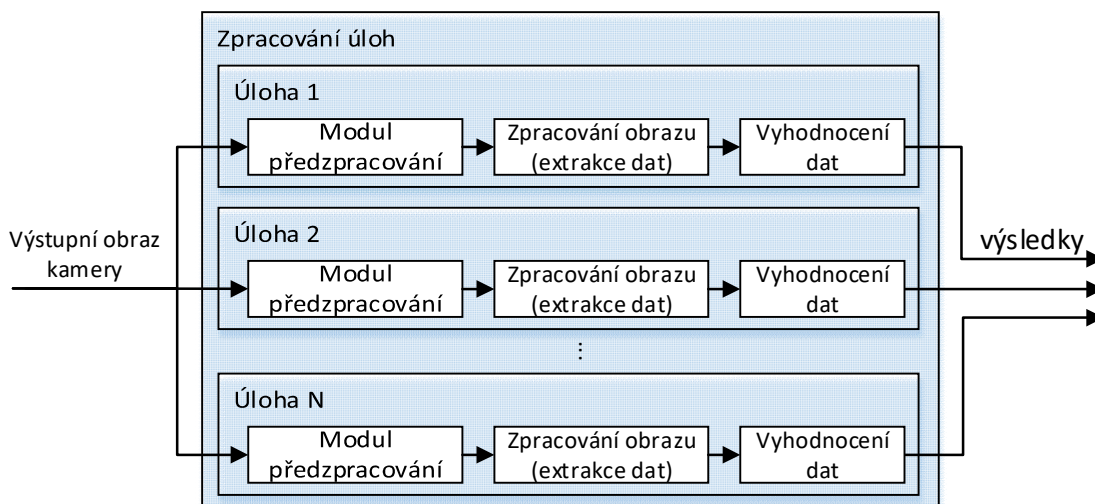
Smyčka obsluhy úloh obsahuje event strukturu pro zachytávání událostí s nastaveným časovým limitem. Událostmi zachytávanými touto event strukturou je především vypršení časového limitu a dále stisknutí tlačítek uživatelského rozhraní. Vypršení časového limitu je běžný stav, kdy není zachycena žádná uživatelská interakce a systém zpracovává všechny předem navolené úlohy. Stisknutí tlačítek vyvolává dialogová okna konfigurace úloh, či kamery a ukládá, či načítá kompletní konfiguraci. Při otevření konfiguračních dialogových oken se vykonávání všech navolených úloh pozastaví do doby, než budou tyto okna zavřena, jelikož se nejedná o asynchronně volané funkce.

Paralelní smyčka obsluhy kamery obsahuje pouze metody pro zpracování příkazů obdržných pomocí fronty (ovládání kamery) a odesílání notifikací.

## Zpracování úloh

Tento úkon, vykonávaný třídou Camera Vision a znázorněný na Obr. 32, se vykonává periodicky v případě, že není spuštěno některé z dialogových oken konfigurace systému. Jak již bylo naznačeno na objektovém modelu systému, všechny úlohy strojového vidění jsou seskupeny v 1D poli Tasks, které je atributem objektu Camera Vision. Zpracování úloh spočívá ve volání virtuální metody Process Image pro každý prvek pole Tasks. Konkrétní spuštěná metoda pak závisí na aktuálním typu objektu neboli úlohy.

Každá úloha alokuje při inicializaci paměť pro vlastní kopii obrazu, nad kterou vykonává potřebné algoritmy. Při spuštění zpracování se sejme kopie z výstupního obrazu objektu Camera a je spuštěno vykonání modulu předzpracování, který nad obrazem vykoná navolenou sekvenci algoritmů. Po dokončení předzpracování se vykoná vlastní algoritmus zpracování a získaná data se zapíše do objektu třídy VIOData. Vyhodnocení dle nastavených parametrů se spustí příslušnou metodou třídy VIOData.



Obr. 32 - Zpracování úloh

## Operační a konfigurační režim zpracování úloh

Metody zpracování obrazu jsou navrženy v souladu s konceptem objektově orientovaného programování, a tedy všechny potřebné parametry získávají z atributů konkrétních objektů. Tím je umožněn běh systému bez uživatelské interakce. Při vytváření úloh jsou vždy nastaveny výchozí hodnoty všech atributů, avšak ty je ve většině případů nutné změnit.

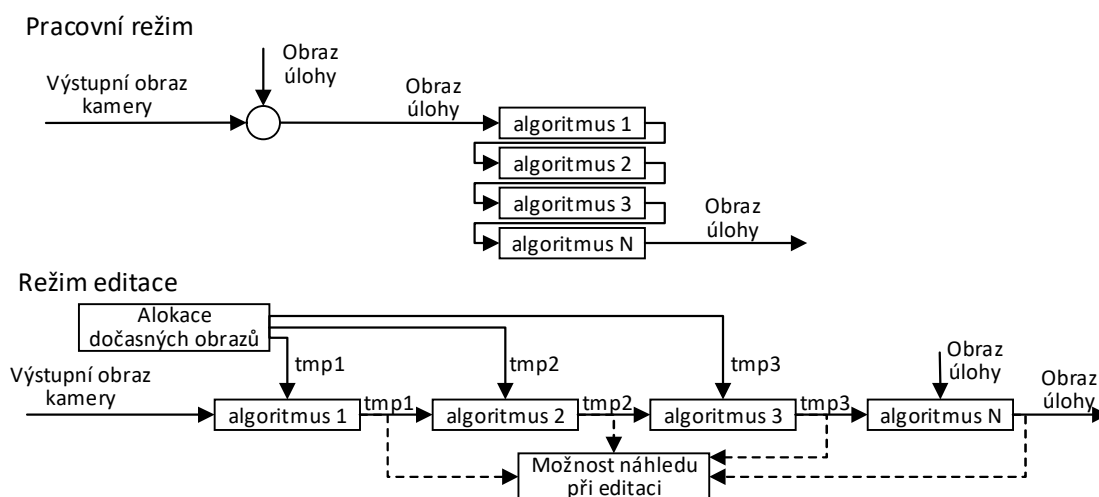
Režimy editace, respektive pozadí konfiguračních oken, byly zpracovány tak, že je vždy volána totožná metoda jako v operačním režimu. Rozdílem však je možnost úpravy všech atributů. Za tímto účelem je vytvořeno prostředí, které z objektu vyčte všechny atributy, načte je do ovládacích prvků a při změně hodnoty některého z nich je zapíše zpět do objektu. Navíc je editační prostředí doplněno o podpůrné funkce, například vykreslení výsledků do obrazu.

## Modul předzpracování obrazu

Tímto modulem je objekt třídy Image Preprocessing a každá úloha disponuje vlastním objektem. Výhodou tohoto uspořádání je možnost konfigurace vlastní sekvence předzpracování pro každou úlohu, dle specifických potřeb. Tím pádem není ovlivněn výstupní obraz kamery, ale pouze obraz dané úlohy.

Sekvence algoritmů, které mají být vykonány, jsou seskupeny v 1D poli Tasks. Podobně jak je tomu u vykonání úloh strojového vidění, předzpracování obrazu spočívá ve volání virtuální metody Preprocess Image pro každý prvek pole Tasks. Konkrétní spuštěná metoda pak závisí na aktuálním typu objektu.

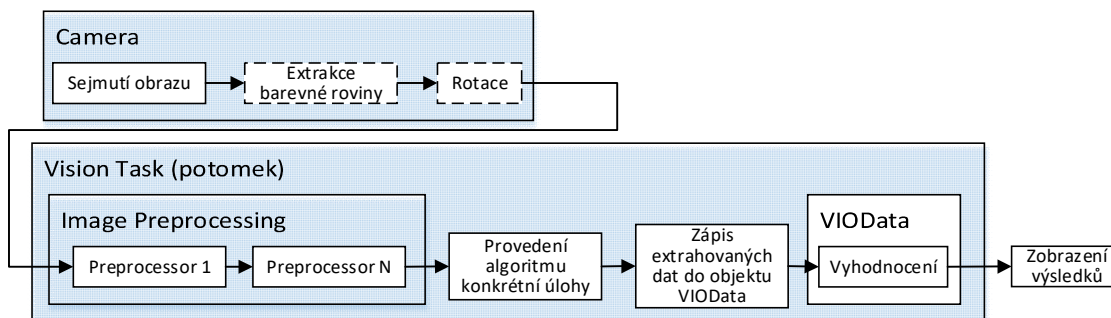
Kromě pracovního režimu, který provádí operace nad stejným obrazem, existuje u modulu předzpracování i režim editace. Tento režim, na rozdíl od pracovního, umožňuje náhled obrazu v jednotlivých stádiích předzpracování. Docíleno je toho alokováním paměti pro několik obrazů, které budou použity pro každý z algoritmů samostatně. Schematické znázornění je na Obr. 33.



Obr. 33 - Pracovní a editační režim modulu předzpracování

## Tok dat

Znázornění sekvence, kterou projdou data od obrazové podoby, přes všechny algoritmy zpracování, až po výsledek je na Obr. 34.

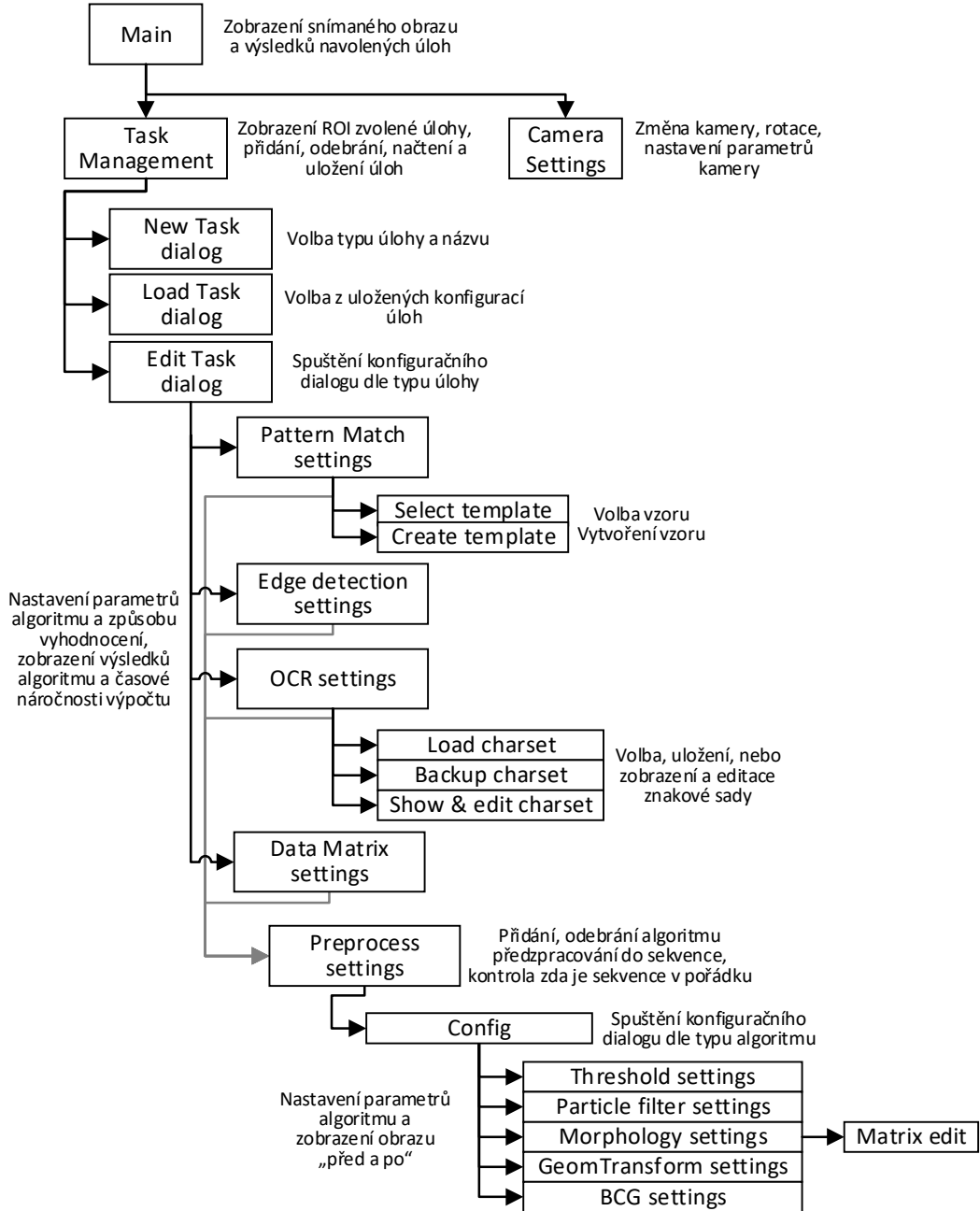


Obr. 34 - Zobrazení toku dat



### 5.4 Uživatelské rozhraní

Základem uživatelského rozhraní je okno Main, které zobrazuje snímaný obraz a výsledky úloh strojového vidění. Z tohoto okna je možné spustit ostatní dialogová okna pro konfiguraci jednotlivých částí systému. Mapa těchto dialogových oken i s jejich stručným popisem je znázorněna na Obr. 35. Většina těchto oken je představena v kapitole 6 - Ukázka využití systému.



Obr. 35 - Mapa dialogových oken s popisem jejich účelu



## 6 UKÁZKA VYUŽITÍ SYSTÉMU

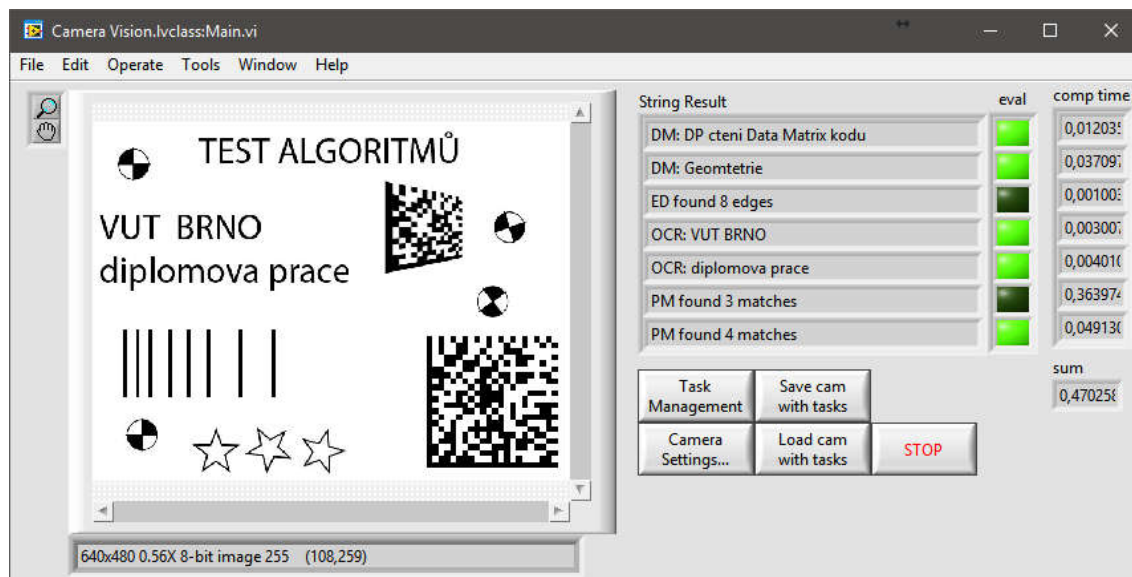
Pro ukázkou algoritmů byl navržen obraz (Obr. 36), na kterém je možná demonstrace všech úloh zpracování. Červeně jsou zobrazeny oblasti zájmu, v kterých jsou zpracovávány jednotlivé úlohy, včetně označení použité úlohy.



Obr. 36 - Testovací obraz algoritmů včetně označení oblastí, nad kterými jsou úlohy prováděny

### Volba kamery a úloh

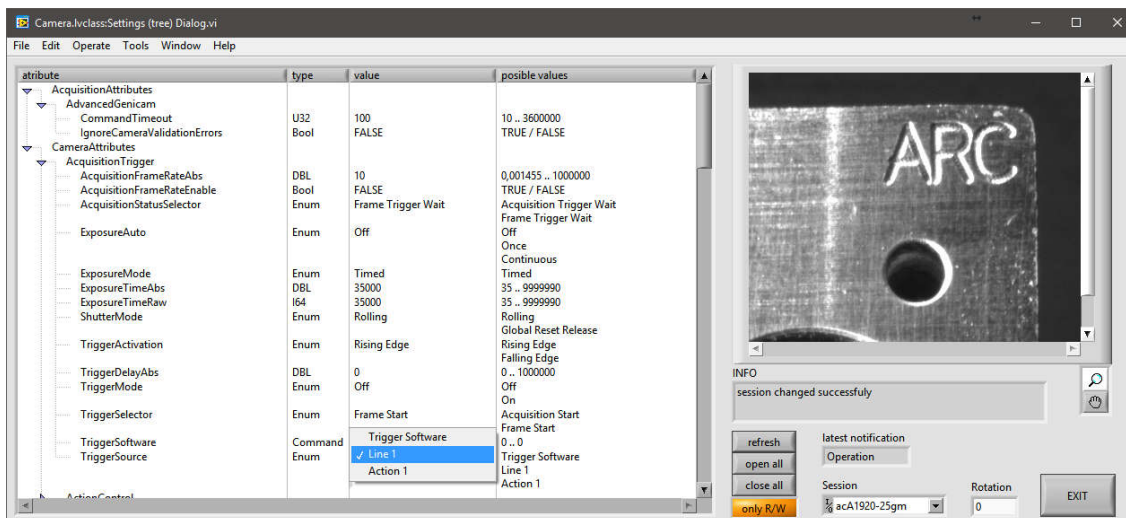
Po spuštění systému se zobrazí hlavní dialogové okno systému. Při prvním spuštění je bez obrazu a je nezbytný výběr kamery a případně její konfigurace, což je možné provést v dialogovém okně nastavení kamery. Při každém dalším spuštění je volba kamery provedena automaticky dle poslední volby. Na Obr. 37 je vyobrazeno hlavní okno systému s již navolenými úlohami.



Obr. 37 - První spuštění systému

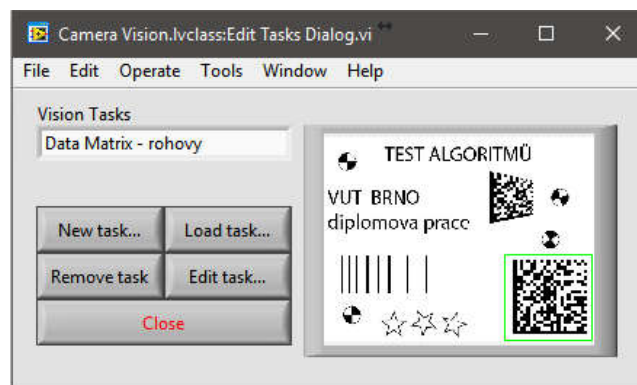
Volba kamery se provádí volbou v rozbalovací nabídce Session, kde se zobrazí všechny dostupné kamery. Po zvolení kamery se vyčtou a zobrazí všechny dostupné atributy. Pro zvýšenou přehlednost byly naprogramovány funkce využívající stromovou strukturu, která umožňuje kategorizaci atributů. Konfigurační okno umožňuje zobrazení

všech atributů, nebo pouze přepisovatelných. Po zvolení atributů se podle datového typu zobrazí ovládací prvek, ve kterém je možné hodnotu atributu změnit. Ukázka dialogového okna konfigurace kamery je na Obr. 38.

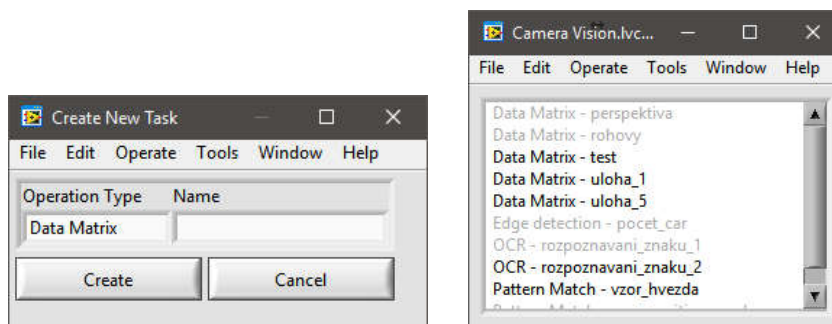


Obr. 38 - Dialogové okno nastavení kamery

Pokud je zvolena kamera, je již možné přidávat a konfigurovat jednotlivé úlohy strojového vidění. Editace úloh je dostupná pod volbou Task Management, která vyvolá dialogové okno zobrazené na Obr. 39. V tomto okně je možné úlohy vytvářet, načítat, ukládat a odstraňovat. Všechny možnosti editace úloh jsou ošetřeny proti vzniku duplicitních názvů úloh, což je možné vidět například na Obr. 40 - vpravo, kde již použité úlohy jsou zašedlé a není možné je zvolit znovu. Při výběru navolené úlohy se zobrazí oblast zájmu, nad kterou je daná úloha zpracovávána.



Obr. 39 - Dialogové okno volby úloh s již navolenými úlohami

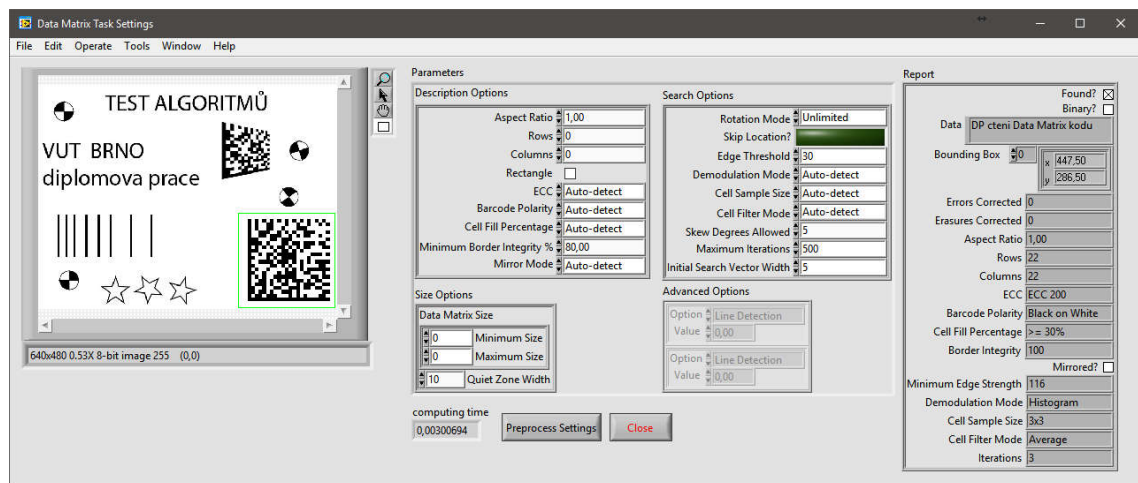


Obr. 40 - Dialogová okna tvorby nové úlohy (vlevo) a načítání uložené úlohy (vpravo)

## Konfigurace úloh

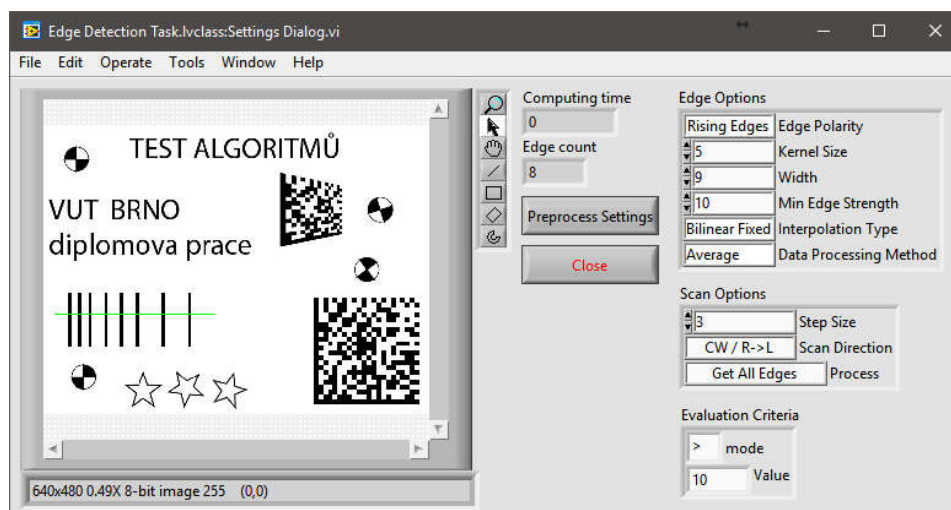
Po navolení úloh strojového vidění je nutné jednotlivé úlohy nakonfigurovat. K tomu slouží dialogová okna, která jsou navržena pro každou úlohu zvlášť. Všechna okna mají však některé společné rysy. Na levé straně je vždy zobrazen právě snímaný obraz a je možné na něm definovat oblast zájmu, nad kterou bude úloha zpracovávána. Pokud není oblast zájmu stanovena, provádí se úloha nad celým obrazem. Dalšími společnými prvky jsou zobrazení výpočtového času dané úlohy, tlačítko pro spuštění konfigurace algoritmu předzpracování a tlačítko pro ukončení dialogového okna. Ostatní prvky jsou již uzpůsobeny potřebám jednotlivých úloh.

Konfigurační okno třídy Data Matrix Task je zobrazeno na (Obr. 41). Ve střední části konfiguračního okna jsou nastavitelné parametry algoritmu. Parametry nemusí být nastaveny, avšak při nastavení upřesňujících parametrů je zpracování několikanásobně rychlejší.



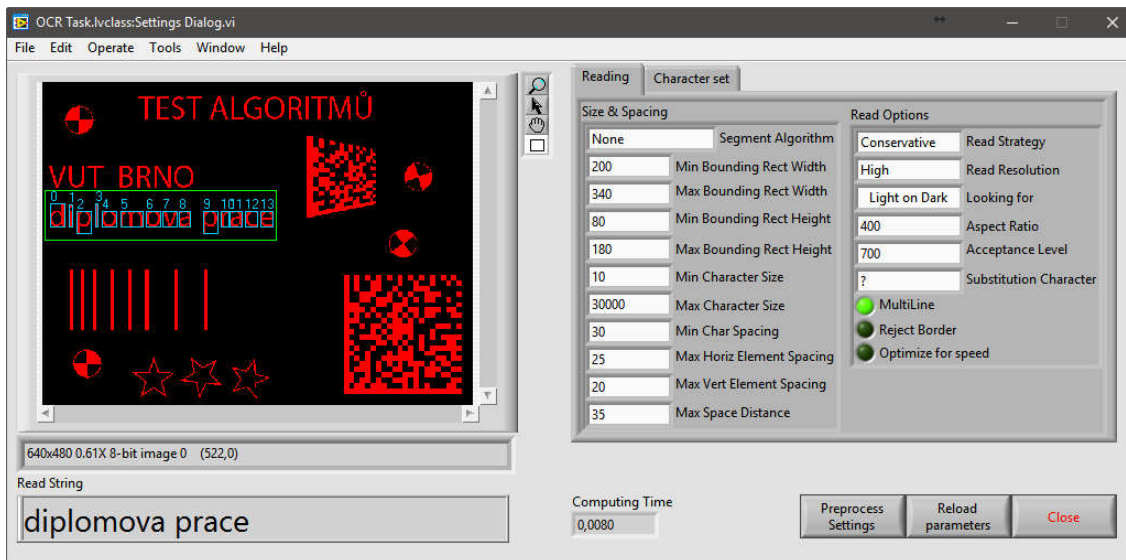
Obr. 41 - Konfigurační okno Data Matrix Task

Třída Edge Detection Task disponuje konfiguračním oknem zobrazeným na (Obr. 42). Podstatnou, změnou oproti ostatním konfiguračním oknům je možnost volby více druhu oblastí zájmů, konkrétně úsečky, obdélníku a mezikruží. V pravé části jsou pak možnosti konfigurace algoritmu detekce hran a podmínky vyhodnocení.



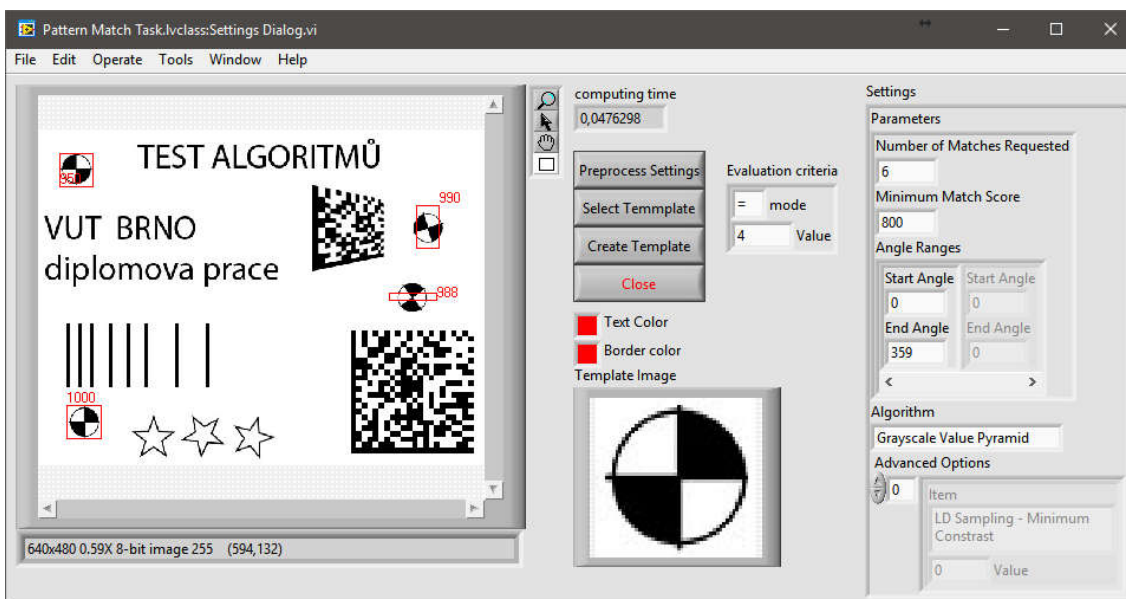
Obr. 42 - Konfigurační okno Edge Detection Task

Podoba konfiguračního okna třídy OCR Task je zobrazena na (Obr. 43). Specifickými prvky tohoto okna je indikátor rozpoznaného textu, který je umístěn pod obrazem a dvou-záložkový ovládací prvek pro nastavení algoritmů. První záložka slouží k nastavení parametrů rozpoznávání znaků a druhá obsahuje možnosti pro vytváření, editaci, ukládání a načítání znakové sady.



Obr. 43 - Konfigurační okno OCR Task

Poslední uvedené konfigurační okno úlohy náleží třídě Pattern Match Task (Obr. 44). V pravé části okna jsou možnosti nastavení algoritmů vyhledávání vzorů. Prostřední část pak obsahuje ovládací prvky vyvolávající další dialogová okna pro volbu, či vytvoření šablony, která bude v obraze hledána. Tato šablona je rovněž zobrazena ve střední části okna. Nalezené výsledky jsou označeny přímo v obraze a je možné zvolit barvu označení.

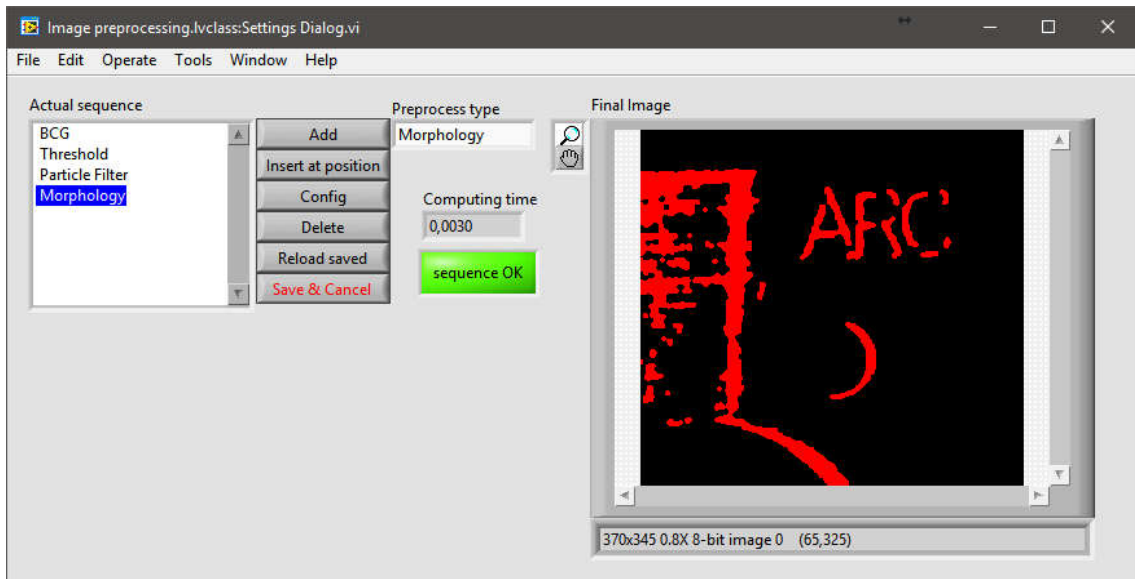


Obr. 44 - Konfigurační okno Pattern Match Task



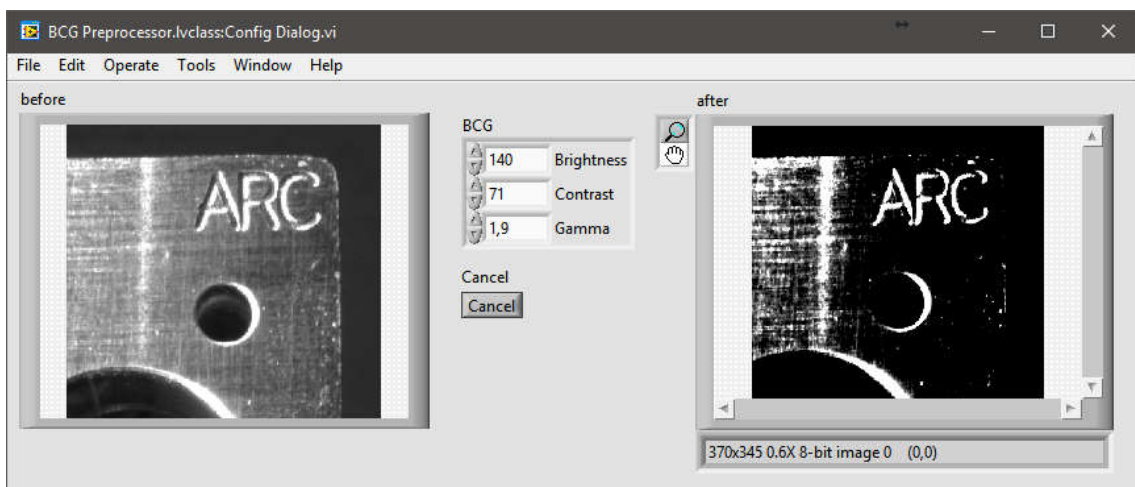
## Konfigurace předzpracování

Hlavní konfigurační okno modulu předzpracování (Obr. 45) slouží k editaci sekvence algoritmů, která je zobrazena v levé části. Pomocí ovládacích prvků je možné přidávat, odebírat, načítat a konfigurovat jednotlivé algoritmy v sekvenci. Při jakékoli úpravě je vždy zkontrolováno, zda je sekvence v pořádku vzhledem k požadované podobě výstupního obrazu. Na uvedeném obrázku je ukázka sekvence předzpracování pro úlohu OCR. Pro ukázkou použití algoritmů předzpracování je použit obraz zachycující kovovou součást a na ní tři vyfrézované znaky.



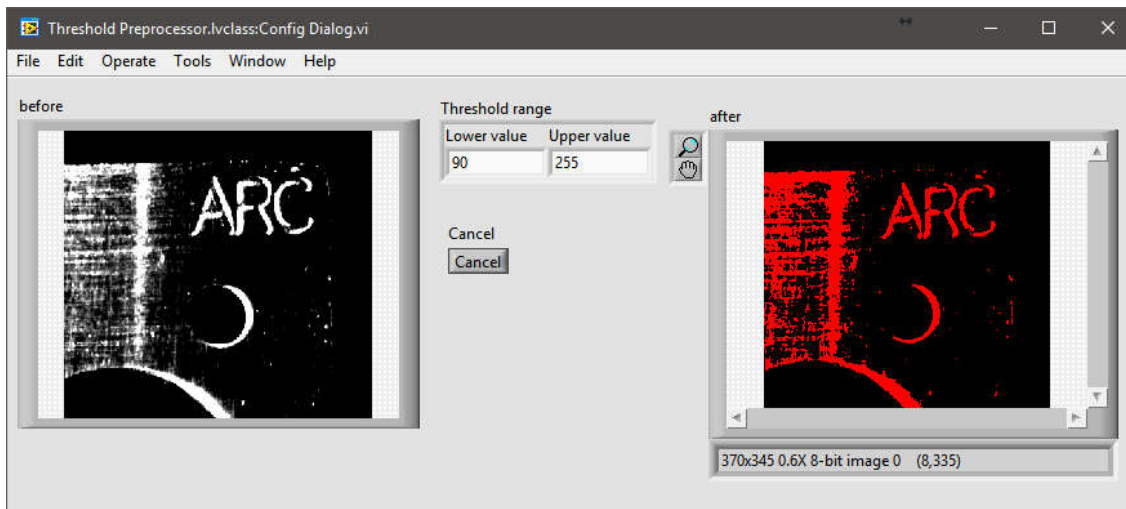
Obr. 45 - Konfigurační okno modulu předzpracování

Prvním použitým algoritmem předzpracování je úprava jasu, kontrastu a gammy. V kontextu předzpracování obrazu pro OCR je vhodné dosáhnout co nejvyššího kontrastu, avšak se zachováním celistvosti čteného textu a potlačením šumu. Ukázka aplikace algoritmu je na Obr. 46.



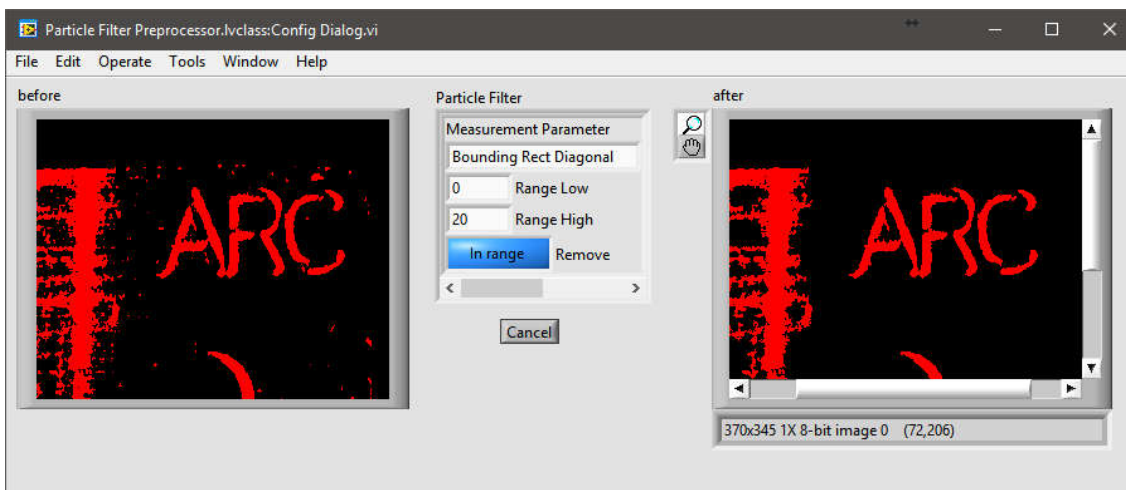
Obr. 46 - Konfigurační okno BCG Preprocessor

Jelikož algoritmus OCR vyžaduje obraz v binární podobě, je nezbytné použít prahování. V konfiguračním okně prahování (Obr. 47) je možné nastavit, jaký rozsah hodnot ze stupnice šedi bude převeden na hodnotu 1. Zbytek bude nabývat hodnoty 0.



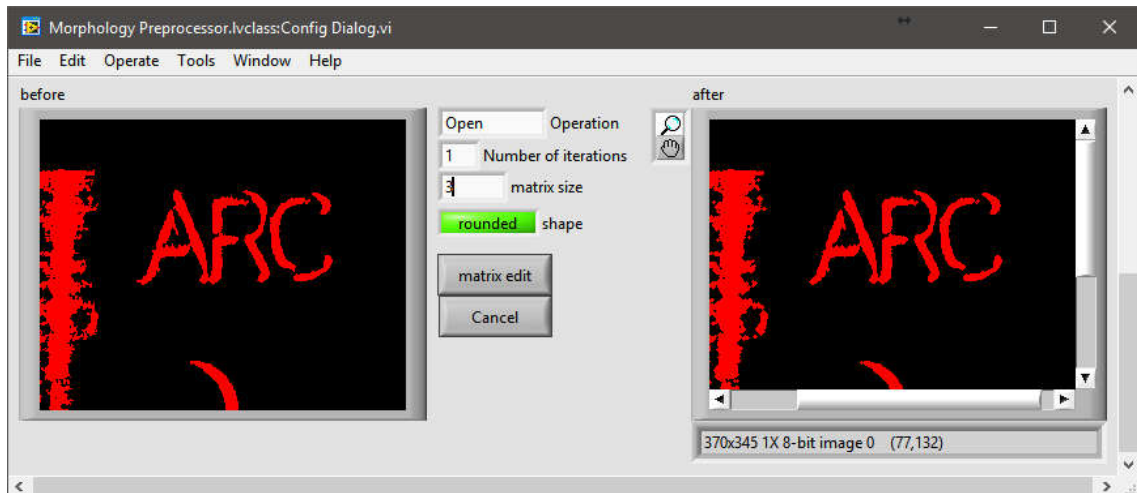
Obr. 47 - Konfigurační okno Threshold Preprocessor

Jak je možné vidět na Obr. 47, výstupní obraz po prahování obsahuje šum, který je nežádoucí. Odstranění šumu je možné provést více způsoby, avšak v tomto případě je využít filtr částic. Nastavení filtru je možné vidět na Obr. 48, kde v prostřední části jsou parametry algoritmu pro vyhodnocení, které částice mají být odstraněny, případně zachovány.



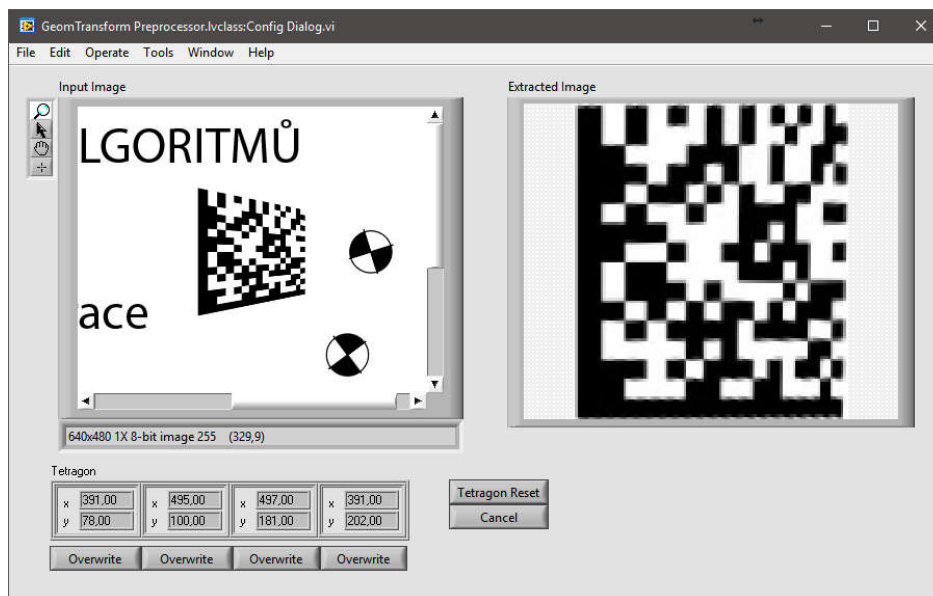
Obr. 48 - Konfigurační okno Particle Filter Preprocessor

Ačkoli již bez větší nutnosti, je možné zařadit do sekvence i algoritmus vykonávající morfologii. Tento algoritmus je možné využít například pro vyhlazení okrajů objektů. Na Obr. 49 je ukázka tohoto využití. Bohužel však v tomto případě není možné použít silnější vyhlazení pomocí morfologie otevření, jelikož by došlo k porušení celistvosti textu. Vhodné využití tohoto algoritmu je například v případě silného zašumění obrazu, kde filtr částic nedokáže odstranit částice šumu spojené s písmem.



Obr. 49 - Konfigurační okno Morphology Preprocessor

Posledním algoritmem předzpracování je geometrická transformace, jejíž konfigurace je zobrazena na Obr. 50. Pro demonstraci funkce algoritmu byl použit jiný obraz, než u předchozích ukávek. Vhodnost aplikace tohoto algoritmu je v případech, jako je vyobrazen na demonstračním obrázku, tedy pokud je potřeba provést korekci perspektivy.



Obr. 50 - Konfigurační okno GeomTransform Preprocessor





## 7 ZÁVĚR

Pro zasazení práce do kontextu problematiky strojového vidění a splnění všech bodů zadání, byla úvodem vypracována rešeršní část. V rešeršní části práce byly popsány části systému strojového vidění jak po stránce hardwarové (osvětlení, objektivy, kamery, rozhraní a výpočetní platforma), tak i softwarové (reprezentace obrazu, základní algoritmy předzpracování a zpracování). Jelikož většina elementárních algoritmů se nepoužívá samostatně, ale v kombinaci s jinými, byla vypracována i kapitola ukazující běžně řešené problémy s popisem použitých algoritmů. Pro vytvoření lepší představy o funkci algoritmů byly vytvořeny i exemplární obrázky, znázorňující jejich aplikaci.

V úvodu praktické části bylo vypracováno pojednání o obecných aspektech návrhu softwarového systému a úrovni přizpůsobitelnosti systému. Přizpůsobitelnost systému je dále rozpracována a je graficky znázorněno, jakých stupňů uživatelské přizpůsobitelnosti je teoreticky možno dosáhnout. Zbytek praktické části je věnován popisu softwarového systému, který byl vyvinut.

Vývoj systému strojového vidění probíhal od základní sekvence postupným rozšiřováním funkcionality. Konečná podoba systému umožňuje volbu kamery připojené pomocí USB a GigE Vision, volbu a konfiguraci několika úloh strojového vidění (současně běžících), plnou konfigurovatelnost sekvence předzpracování pro každou úlohu samostatně a základní konfiguraci vyhodnocení každé úlohy. Ačkoli je většina popisu práce věnována konfiguraci systému, hlavní vlastností systému je schopnost provádění všech nakonfigurovaných algoritmů bez zásahu obsluhy.

Při vývoji systému strojového vidění bylo použito objektově orientované programování. Návrh systému je konceptuálně postaven tak, že je možné jej jednoduchým způsobem rozšířit o další úlohy strojového vidění, či algoritmy předzpracování. Rozšíření by tedy představovalo pouze vytvoření děděné třídy a implementaci jejich metod, bez nutnosti úpravy řídicích částí systému. Software je ve stádiu plné funkčnosti, s ošetřením výjimek a uživatelských vstupů.

Další vývoj systému by mohl být veden ve směru rozšiřování nabídky úloh strojového vidění a algoritmů předzpracování, nebo rozšířením přizpůsobitelnosti. Největší změnou přizpůsobitelnosti by byla unifikace tříd algoritmů předzpracování a zpracování. Výsledkem této změny by byla možnost tvorby rozvětvených sekvencí algoritmů, kde by mohly být zřetězeny i algoritmy zpracování. Prakticky vzato by tedy mohly data extrahovaná z obrazu definovat parametry následujících algoritmů.



## 8 SEZNAM POUŽITÉ LITERATURY

- [1] HORNBERG, Alexander. *Handbook of machine vision*. Wiley-VCH, 2006. ISBN 978-3-527-40584-8.
- [2] HAVLE, Otto. Strojové vidění IV: Osvětlovače. *Automa: časopis pro automatizační techniku*. b.r., 2008(04), 47-49. Dostupné také z: [http://automa.cz/Aton/FileRepository/pdf\\_articles/36988.pdf](http://automa.cz/Aton/FileRepository/pdf_articles/36988.pdf)
- [3] Objektiv. *Megapixel* [online]. b.r. [cit. 2017-03-14]. Dostupné z: <https://www.megapixel.cz/objektiv>
- [4] MOTYČKA, Jakub. KOUKAAM A.S. *CCTV objektivy*. 2007. Dostupné také z: [http://www.koukaam.se/koukaam/downloads/MAN\\_CZ\\_Objektivy\\_uzivatelsky\\_manual.pdf](http://www.koukaam.se/koukaam/downloads/MAN_CZ_Objektivy_uzivatelsky_manual.pdf)
- [5] The Advantages of Telecentricity. *Edmund optics: 75 years of optics* [online]. b.r. [cit. 2017-03-14]. Dostupné z: <https://www.edmundoptics.com/resources/application-notes/imaging/advantages-of-telecentricity/>
- [6] White Papers: Choosing the Right Camera Bus. *National Instruments* [online]. 2016 [cit. 2017-03-20]. Dostupné z: <http://www.ni.com/white-paper/5386/en/>
- [7] SEDLÁŘ, Martin, Jaromír ŠRÁEK, Vojtěch MORNSTEIN a Ondřej MORNSTEIN. *Získávání a analýza obrazové informace*. 2011. b.r. Dostupné také z: <https://www.med.muni.cz/biofyz/Image/ucebnice.pdf>
- [8] DEMANT, Christian, Bernd STREICHER-ABEL a Peter WASZKEWITZ. *Industrial image processing: visual quality control in manufacturing*. New York: Springer, 1999. ISBN 35-406-6410-6.
- [9] SONKA, Milan, Vaclav HLAVAC a Roger BOYLE. *Image Processing, Analysis, and Machine Vision*. Third Edition. Thomson Learning, 2008. ISBN 978-0-495-24428-7.
- [10] EUROPEAN MACHINE VISION ASSOCIATION (EMVA). *GenICam standard*. b.r. Version 2.0. Dostupné také z: [http://www.emva.org/wp-content/uploads/GenICam\\_Standard\\_v2\\_0.pdf](http://www.emva.org/wp-content/uploads/GenICam_Standard_v2_0.pdf)



## 9 SEZNAM OBRÁZKŮ

Obr. 1 - Některé interakce komponent systému strojového vidění. (pouze výběr) [1]...	17
Obr. 2 - Porovnání různé geometrie osvětlení součásti s lesklým povrchem a vyraženým logem, vlevo bright-field, vpravo dark-field. [2].....	18
Obr. 3 - Modrý potenciometr osvětlen modře (vlevo) a žlutě (vpravo), snímán černobílou kamerou. [1].....	19
Obr. 4 - Geometrické zkresení (originál, poduškovitost, soudkovitost).....	21
Obr. 5 - Uspořádání scény, snímek běžným objektivem, snímek telecentrickým objektivem [5].....	21
Obr. 6 - Použití telecentrického objektivu pro měřicí úlohy (vlevo klasický, vpravo telec. objektiv) [5] .....	21
Obr. 7 - HSL a HSV barevné modely, autor: Jacob Rus .....	27
Obr. 8 - Převod barevného obrazu do stupnice šedi .....	28
Obr. 9 - Prahování.....	28
Obr. 10 - Ukázka zvýšení jasu a kontrastu .....	29
Obr. 11 - Princip lineárního filtru. ....	30
Obr. 12 - Ukázka lineárních filtrů pro rozmazání, včetně aplikovaných jader.....	30
Obr. 13 - Ukázka lineárních filtrů pro detekci hran, včetně aplikovaných jader.....	30
Obr. 14 - Sobbelova a Cannyho detekce hran .....	31
Obr. 15 - Segmentace pomocí prahování - výsledkem jsou čtyři segmenty.....	32
Obr. 16 - Houghova transformace - detekce kružnic .....	33
Obr. 17 - Watershed segmentace .....	34
Obr. 18 - Detekce vzoru - nalezení polohy šroubů .....	35
Obr. 19 - Čtení 2D kódu (QR) .....	37
Obr. 20 - OCR - kovový frézovaný povrch s frézovaným značením .....	38
Obr. 21 - Detekce povrchových vad pomocí detekce hran - održený kousek materiálu při frézování .....	40
Obr. 22- Detekce objektů - konektory .....	41
Obr. 23 - Různé zřetězení sekvence algoritmů .....	44
Obr. 24 - Systém strojového vidění – sekvence.....	46
Obr. 25 - Systém SV - výběr několika úloh.....	46
Obr. 26 - Systém SV - několika úloh s modulem předzpracování .....	47
Obr. 27 - Systém SV - jedna sekvence předzpracování, několik úloh.....	47
Obr. 28 - Systém SV s možností tvorby závislosti algoritmů a modulem výpočtů a vyhodnocení.....	48
Obr. 29 - Systém SV - libovolné zřetězení algoritmů.....	48
Obr. 30 - Zjednodušený objektový model systému .....	50
Obr. 31 - Hlavní řídicí mechanismus.....	53
Obr. 32 - Zpracování úloh.....	54
Obr. 33 - Pracovní a editační režim modulu předzpracování .....	55
Obr. 34 - Zobrazení toku dat.....	55

Obr. 35 - Mapa dialogových oken s popisem jejich účelu .....	56
Obr. 36 - Testovací obraz algoritmů včetně označení oblastí, nad kterými jsou úlohy prováděny .....	57
Obr. 37 - První spuštění systému .....	57
Obr. 38 - Dialogové okno nastavení kamery .....	58
Obr. 39 - Dialogové okno volby úloh s již navolenými úlohami .....	58
Obr. 40 - Dialogová okna tvorby nové úlohy (vlevo) a načítání uložené úlohy (vpravo) .....	58
Obr. 41 - Konfigurační okno Data Matrix Task .....	59
Obr. 42 - Konfigurační okno Edge Detection Task .....	59
Obr. 43 - Konfigurační okno OCR Task .....	60
Obr. 44 - Konfigurační okno Pattern Match Task .....	60
Obr. 45 - Konfigurační okno modulu předzpracování .....	61
Obr. 46 - Konfigurační okno BCG Preprocessor .....	61
Obr. 47 - Konfigurační okno Threshold Preprocessor .....	62
Obr. 48 - Konfigurační okno Particle Filter Preprocessor .....	62
Obr. 49 - Konfigurační okno Morphology Preprocessor .....	63
Obr. 50 - Konfigurační okno GeomTransform Preprocessor .....	63

## **10 SEZNAM PŘÍLOH**

CD s elektronickou verzí práce