

Studentská tvůrčí a odborná činnost
STOČ 2011

**ŘÍZENÍ POLOHOVÁNÍ NÁSTROJE PRŮMYSLOVÉHO
ROBOTA STÄUBLI UNIMATION TX-40 POMOCÍ SNÍMÁNÍ
SVĚTELNÉHO OBRAZCE**

Bc. Tomáš GAVENDA

Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, Nad Stráněmi
4511, 760 05 Zlín, Česká republika

5. května 2011
VŠB-TU v Ostravě

Klíčová slova: průmyslový robot, Stäubli Unimation TX-40, algoritmy pro vyhledávání přímek, sériová komunikace, VAL3, inteligentní kamera CMUcam3

Anotace: Účelem práce je navrhnout efektivní metodu pro vyhledání světelného obrazce a jeho následné zpracování pro řízení polohování nástroje robota. Inteligentní kamera je společně s laserem umístěna na nástroji robota a snímá strukturovaný světelný obrazec promítaný laserem na rovinu v pracovním prostoru robota. Robot je řízen mikropočítačem inteligentní kamery, která komunikuje přes sériový port s řídicí jednotkou robota. V mikropočítači kamery probíhá zpracování algoritmů pro vyhledání strukturovaného světelného obrazce, určení jeho parametrů a výpočtu údajů sloužících pro vydání pokynů k polohování nástroje. Pro programování mikropočítače je užit jazyk C, robot je naprogramován v jazyku VAL3, což je programovací jazyk společnosti Stäubli.

Obsah

1.	Úvod a zadání práce	3
2.	Laser a strukturovaný světelný obrazec	3
3.	Výběr inteligentní kamery a její programování.....	3
4.	Algoritmy pro vyhledání bodů a následné určení přímek.....	4
4.1	Nastavení barevného prostoru a citlivosti kamery	4
4.2	Algoritmus vyhledání bodů	5
4.3	Algoritmy a metody pro sestrojení přímek	6
4.4	Zpracování přímek pro informace k polohování.....	8
5.	Průmyslový robot Stäubli Unimation TX-40 a jeho programování.....	9
6.	Spojení inteligentní kamery a průmyslového robota	9
7.	Závěr	10
	Literatura.....	10

1. Úvod a zadání práce

Předem bych rád poděkoval panu Ing. et Ing. Eriku Královi za uvedení do problematiky a celkovou pomoc během práce a panu Ing. Petru Navrátilovi PhD. za cenné rady ohledně nastavení průmyslového robota Stäubli.

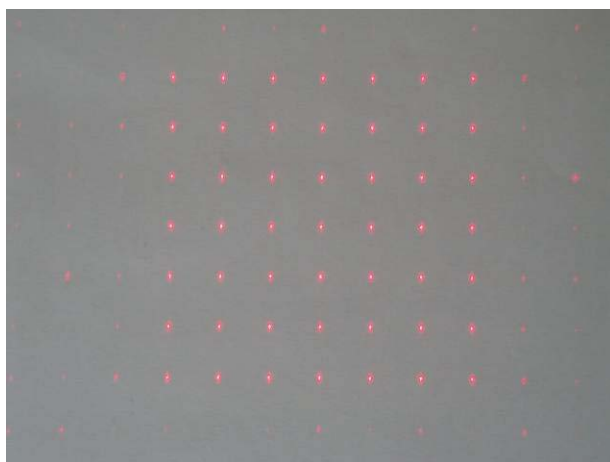
Tato práce slouží je aplikovatelná v rámci různých odvětví průmyslu (bezpečnostní, strojírenský, automobilový, stavební, apod.) a přináší předběžné výsledky, kterých jsem během tohoto akademického roku jako pomocná vědecká síla dosáhl.

Zadáním bylo navrhnout a zrealizovat efektivní metodu pro vyhledání a popsání strukturovaného světelného obrazce, promítaného laserem na rovinu v pracovním prostoru robota a snímaném inteligentní kamerou. Informace vytvořené těmito metodami jsou pak použity pro řízení polohování nástroje robota, na kterém je laser a inteligentní kamera umístěny.

2. Laser a strukturovaný světelný obrazec

Použitý laser září paprsky ve viditelném spektru o vlnové délce 660 nm, což odpovídá červené barvě. Výkon laseru je poměrně malý – 20 mW, ale pro účely práce se ukázalo, že dostatečný.

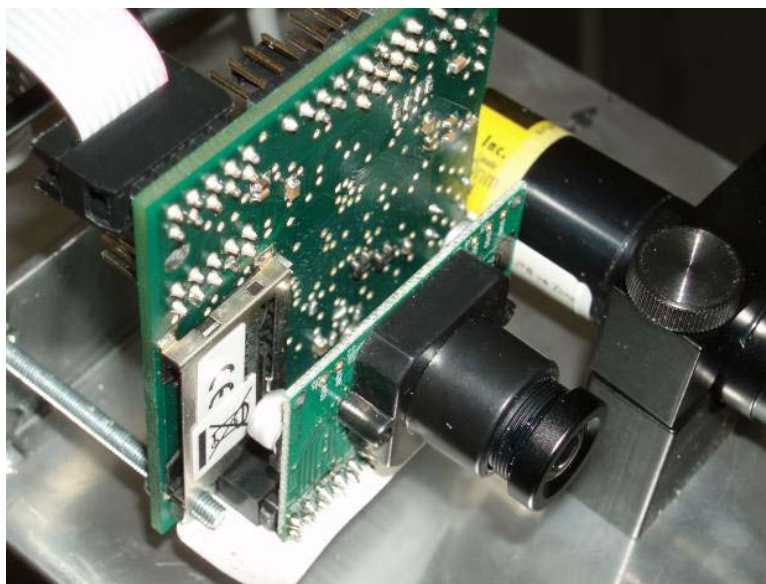
Strukturovaný světelný obrazec je tvořen optickou mřížkou, vloženou do optiky laseru. Tato optická (difrakční) mřížka pak vytváří hlavní matici bodů – maxim – o rozměrech 7x7 bodů. Minima tvořená touto mřížkou jsou slabá a při správném nastavení kamery zanikají.



Obr. 1: Strukturovaný světelný obrazec

3. Výběr inteligentní kamery a její programování

Pro účely práce byla vybrána inteligentní kamera CMUcam3, která se vyznačuje především svou nízkou cenou, kompaktností, nízkými požadavky na napájení a především tím, že je „open source“, to znamená, že její zdrojové kódy jsou volně dostupné a modifikovatelné. Navíc lze mikropočítač (s označením Philips LPC2106) této inteligentní kamery programovat v jazyce C (resp. C99).



Obr. 2: Inteligentní kamera CMUcam3.

Jako překladač slouží kompilátor ARM GCC, který je pod operačním systémem Windows nutné spouštět pomocí vývojového prostředí Cygwin. Pro nahrání přeloženého programu z PC do mikropočítače kamery se využívá sériového rozhraní a LPC210x Flash Utility od společnosti Philips. Pro testování kamery a výpis dat, která kamera odesílá, se využívá programu CMUcam3 Frame Grabber. [1]

4. Algoritmy pro vyhledání bodů a následné určení přímk

Po výběru kamery následuje určení metod pro vyhledávání bodu a výběr efektivní metody pro sestrojení přímek, určujících hranice strukturovaného obrazce a následně vrcholy čtyřúhelníka, pomocí kterého je možné polohovat nástroj robota v prostoru.

4.1 Nastavení barevného prostoru a citlivosti kamery

Pro správné vyhledání bodů, ze kterých je strukturovaný obrazec tvořen, bylo nutné nastavit správný barevný prostor – barevný režim kamery. Použitá inteligentní kamera má k dispozici tři barevné prostory – RGB, $Y C_R C_B$ a HSV. Následující tabulka předvádí několik možných nastavení:

Jas: 0 Kontrast: 0 Prostor: RGB		Jas: 0 Kontrast: 0 Prostor: YC _R C _B	
Jas: 50 Kontrast: 50 Prostor: RGB		Jas: 100 Kontrast: 100 Prostor: YC _R C _B	
Jas: 255 Kontrast: 100 Prostor: RGB		Jas: 255 Kontrast: 100 Prostor: YC _R C _B	

Tab. 1: Porovnání snímků z kamery při různém nastavení jasu, kontrastu a barevného prostoru.

Z dalšího testování nakonec vyplynulo, že CMOS snímač inteligentní kamery vnímá příliš jasnou červenou barvu laseru spíše jako bílou, a proto byly vyřazeny prostory RGB a YC_RC_B a byl upřednostněn barevný prostor HSV, jelikož jeho složka *V* udává přímou hodnotu intenzity jasu a je vhodná k dalšímu zpracování. Veškerá nastavení kamery se programují přímo v kódu programu přes implementované knihovny `cc3.h` a `cc3_ilp.h`. Ukázka kódu:

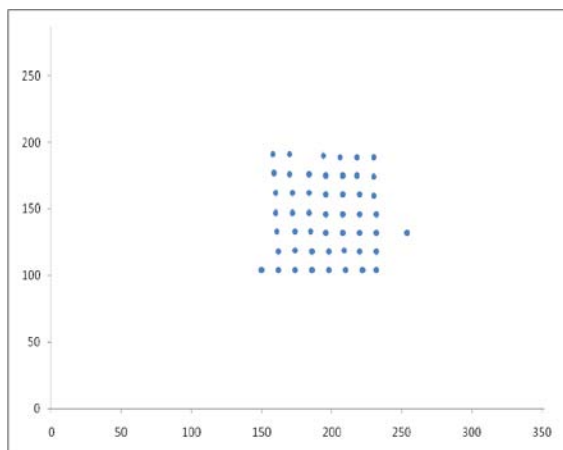
```
cc3_camera_set_colorspace (CC3_COLORSPACE_HSV); //barevný prostor
cc3_camera_set_resolution (CC3_CAMERA_RESOLUTION_HIGH); //rozlišení
cc3_camera_set_auto_white_balance (false); //vyvazování bílé
cc3_camera_set_auto_exposure (false); //automatická expozice
cc3_camera_set_brightness (0); //nastavení jasu
cc3_camera_set_contrast (100); //nastavení kontrastu
```

4.2 Algoritmus vyhledání bodů

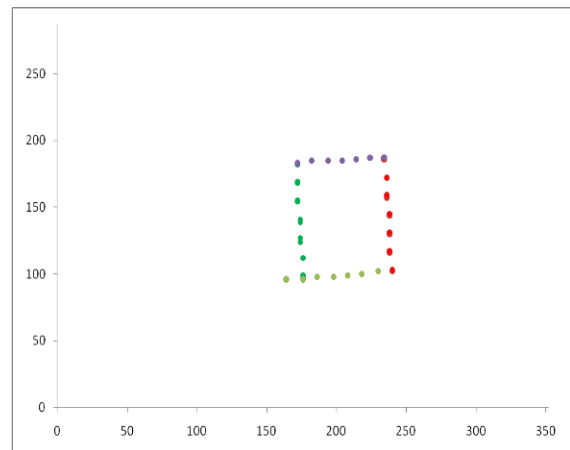
V rámci práce bylo testováno několik metod výběru bodů, které tvoří strukturovaný světelný obrazec. Všechny tyto metody jsou založeny na průchodu snímkem nahraného v paměti kamery, řádek po řádku, pixel po pixelu, a nalezení pixelů s vyšší hodnotou složky *V*, než je hraniční hodnota. Tato hraniční hodnota je dynamicky měněna s každým snímkem, aby byl nalezen dostatečný počet bodů a zároveň aby nebyl nalezen nadbytečný počet bodů. Ukázka kódu průchodu snímkem:

```
cc3_pixbuf_frame_set_coi(CC3_CHANNEL_VAL); //pro urychlení pruchodu se  
bere pouze slozka V  
  
cc3_pixbuf_load (); //nacteni snimku do bufferu  
while (cc3_pixbuf_read_rows (img.pix, 1)) //pruchod snimky po radcich  
{  
    for (uint16_t x = 0; x < img.width; x++) //pruchod radku po pixelech  
    {  
        uint8_t val = ((uint8_t *) img.pix)[x]; //slozka V pixelu  
        if (val > mezjasu) //porovnaní s hranicni hodnotou  
        {  
            ...  
        }  
    }  
    y++; //dalsi radek  
}
```

Finální metody určení bodů jsou pak dvě – při čemž jedna přesně vybírá všechny body strukturovaného světelného obrazce a druhá vybírá okrajové body, rozdělené na čtyři skupiny bodů – první body zleva, zprava, shora a zdola strukturovaného světelného obrazce. Obě pak výsledné body předávají dál na zpracování algoritmům pro sestrojení přímek.



Graf 1: Metoda vybírající všechny body



Graf 2: Metoda vybírající okrajové body.

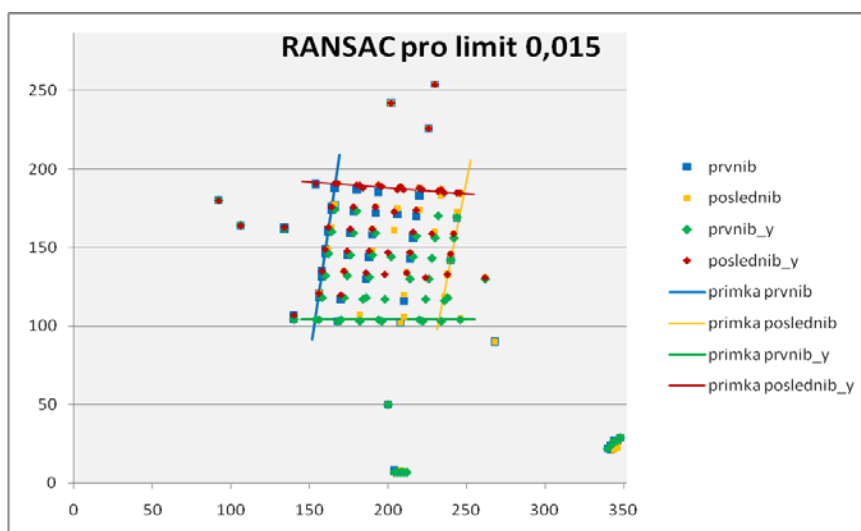
4.3 Algoritmy a metody pro sestrojení přímek

V literatuře zabývající se zpracováním obrazové informace – vyhledáním přímek (např. [3]) se vyskytuje celá řada algoritmů pro vyhledání a sestrojení přímek. Jako nejefektivnější algoritmy v rámci této práce byly, jako základ různých metod, vybrány algoritmy lineární regrese, RANSAC a upravený inkrementální algoritmus. [3]

Ještě před uplatněním algoritmů pro sestrojení přímek, je nutné vybrat vhodný tvar rovnice přímky. Jako nejvhodnější byl zvolen směrnicový tvar přímky $y = ax + b$, protože z něj snadno jedinou matematickou operací odvodíme úhel přímky od osy x rovinného souřadnicového systému, přímka zadaná ve směrnicovém tvaru se snadno zobrazuje a výpočet rovnice přímky ve směrnicovém tvaru ze dvou bodů je také výpočetně nenáročný. Jediný závažný problém je však s přímkami, které jsou kolmé na osu x, protože jejich směrnice a se blíží nekonečnu. Tento případ je však v algoritmech ošetřen, a to tak, že pokud je nalezena kolmá přímka (souřadnice x dvou bodů jsou shodné), udá se jako směrnice a vysoká hodnota (v řádech tisíců). Přímka s vysokou hodnotou a má pak úhel od osy x téměř totožný s 90° (např. $a = 5000 \rightarrow \arctan(a) = 89,989^\circ$).

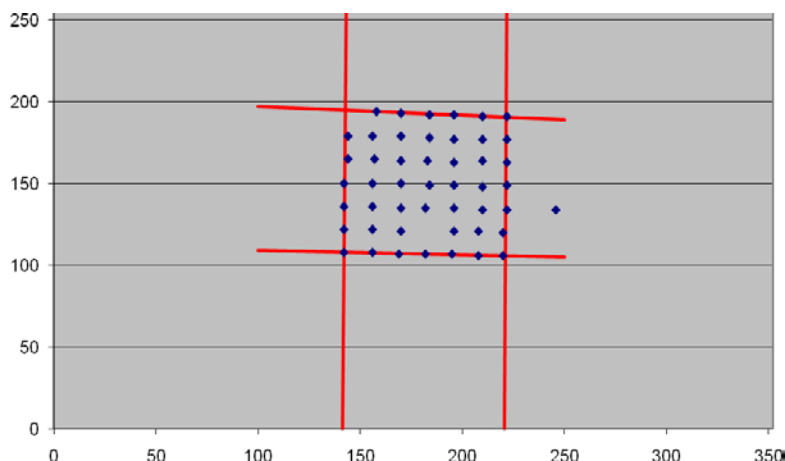
Nejjednodušší algoritmus a metoda sestrojení požadovaných přímek spočívá ve výpočtu přímky metodou nejmenších čtverců z množiny všech bodů, o kterých předpokládáme, že leží na jedné přímce. Metoda nejmenších čtverců je ekvivalentní s lineární regresí – množinou bodů prokládáme lineární závislost $y = ax + b$. Principem metody je nalezení takové přímky, jejíž součet čtverců vzdálenosti od všech bodů v množině je minimální. [2]

Další metodou je metoda využívající algoritmus RANSAC. Ten spočívá v náhodném výběru dvou bodů z množiny a výpočtu přímky, kterou tyto body tvoří. Poté, co je přímka sestrojena, projde se zbytek množiny bodů a vypočítá se jejich vzdálenost. Jeli tato vzdálenost v určitém rozsahu, započítají se tyto body do přímky a algoritmus pokračuje. Po dosažení maximálního počtu iterací následuje vyhodnocení a přímky s nejvyšším počtem bodů jsou vyhodnoceny jako výsledné. Velkou výhodou tohoto algoritmu oproti lineární regresi je, že vůbec nezapočítává do přímek body, které jsou ve velké vzdálenosti od těchto přímek, a proto je přesnější i při výskytu velkého množství falešných bodů. [3]



Graf 3: Výsledné přímky sestrojné pomocí algoritmu RANSAC.

Poslední použitá metoda sestrojuje a vybírá přímky pomocí upraveného inkrementálního algoritmu. Inkrementální algoritmus bez úpravy spočívá v sestrojení přímky z prvních dvou nalezených bodů a postupném průchodu množiny bodů, při čemž pokud následující bod spadá na přímku, je k přímce připočten, nespadá-li, je vytvořena nová přímka a průchod pokračuje obdobným způsobem. [3] Vyvinutá vlastní metoda spočívá ve vypočtení přímky, následném průchodu celé množiny bodů a případném přiřazení bodů a dále pak ve vypočtení dalších přímek, u kterých se ověřuje, zdali již nejsou mezi předchozími přímkami obsaženy, a pro které je opět celá množina bodů prohledána a jednotlivé body jsou přiřazeny. Na závěr metody jsou vybrány přímky s nejvíce body a z nich pak okrajové přímky – to znamená přímky tvořící čtyřúhelník (strukturovaný světelný obrazec).

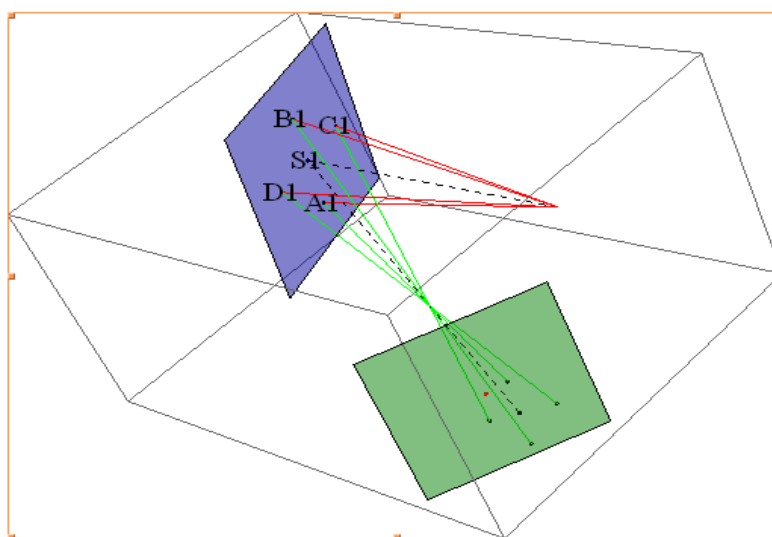


Graf 4: Výsledné přímky sestrojené pomocí pokročilého inkrementálního algoritmu.

4.4 Zpracování přímek pro informace k polohování

Poslední krok výpočtů v inteligentní kameře je zpracování údajů o přímkách do takové podoby, aby mohly být odeslány do řídicí jednotky robota. V současné verzi robot zpracovává pouze informace o rotaci nástroje, ale v následujících verzích bude ještě přidána schopnost měnit pracovní výšku. Požadovaná poloha nástroje je poloha kolmá k rovině, na kterou je promítán strukturovaný světelný obrazec.

Informace o rotacích nástroje robota vycházejí z vypočteného poměru stran čtyřúhelníku, jehož vrcholy A, B, C a D leží v jednotlivých průsečících okrajových přímek. Ze souřadnic těchto bodů je vypočtena délka jednotlivých stran čtyřúhelníku a následně jsou vypočteny poměry protějších stran AB ku CD a BC ku DA, které udávají, na kterou stranu se má nástroj robota naklánět. Výsledek je odeslán do řídicí jednotky robota ve formě řetězce „náklon x+, náklon x-, náklon y+, náklon y-“, kde jednotlivé náklony mohou mít hodnotu 0 či 1, podle toho, na kterou stranu se má robot naklánět (v které ose má rotovat).



Obr. 4: Znáznornění matematického modelu.

5. Průmyslový robot Stäubli Unimation TX-40 a jeho programování

Průmyslový robot Stäubli Unimation TX-40 je kloubový robot pracující v šesti stupních volnosti, má vlastní souřadnicový systém, jehož počátek a osy lze definovat kdekoli v jeho pracovním prostoru. Poté je možno pohybovat nástrojem přesně podle požadavků uživatele. Programování robota se provádí v jazyku VAL3, ve vývojovém prostředí VAL3 Studio, což je software společnosti Stäubli Faverges vyvinutý speciálně pro jejich průmyslové roboty, či přímo přes ovládací panel robota. [4]



Obr. 5: Ovládací panel robota Stäubli TX-40.

Jazyk VAL3 má jednoduchou syntaxi podobnou například PASCALu. Důležité je však zvolit správné funkce pro úpravu bodu, do kterého se má robot pohnout (funkce `appro()`) a pro samotný pohyb (funkce `movej()`). Ještě před samotným programováním je třeba definovat nástroj vůči přírubě – jeho posun a rotaci a následně pak souřadnicový systém, ve kterém se má pohybovat. Toto se provádí pomocí teaching módu, ve kterém je možné robota přes ovládací panel „naučit“ potřebné informace.

Údaje, které robot potřebuje pro polohování nástroje, přijímá řídicí jednotka ze sériového rozhraní, kterým je připojena inteligentní kamera. Data ve formě řetězce znaků jsou převedena na jednotlivé znaky, které jsou vyhodnoceny a podle nichž se upraví souřadnice bodu, do kterého se má nástroj přesunout.

6. Spojení inteligentní kamery a průmyslového robota

Komunikace inteligentní kamery a řídicí jednotky robota probíhá přes sériové rozhraní. Na obou stranách je nutno nastavit sériové porty. Na straně inteligentní kamery pomocí následujícího příkazu:

```
cc3_uart_init(0,CC3_UART_RATE_115200,CC3_UART_MODE_8N1,CC3_UART_BINMODE_T  
EXT);
```

Na straně robota jsou potřeba přes ovládací panel nastavit stejné hodnoty baud rate, parity, počtu datových bitů a režimu a poté je nutné propojit lokální sériovou proměnnou programu se sériovým vstupem pomocí příkazu `sioLink(sio,io:portSerial1)`, kde `sio` je sériová proměnná a `io:portSerial1` sériový port.

Komunikace pak probíhá v cyklu hlavního programu, data odeslaná z inteligentní kamery se ukládají do bufferu řídicí jednotky a jsou oddělena ukončovacím znakem. V této práci byl jako ukončovací znak zvolen @, ale je možné zvolit znak libovolný. Jako ochrana proti přetečení dat v bufferu, slouží funkce v jazyku VAL3 `clearBuffer()`, spouštěná v programu před načítáním dat.

7. Závěr

V této práci jsou obsaženy pouze stručné postupy práce a výsledky, protože byl stanoven limit stránek práce. Pro komplexní vysvětlení problematiky je nejlepší praktická ukázka práce s robotem a procházení zdrojových kódů programů.

Zadání práce bylo splněno navržením a realizací vhodné metody pro zpracování obrazové informace z kamery a následné polohování nástroje robota. Bylo dosaženo propojení inteligentní kamery s robotem a vytvoření funkční komunikace, a následného řízení průmyslového robota pouze za pomoci dat, které jsou automaticky z inteligentní kamery zasílány.

Následující zpracovávání a vylepšování této práce, v rámci diplomové práce, bude rozvinutí její aplikace v bezpečnostním průmyslu, což je jeden z cílů zmíněné diplomové práce.

Literatura

- [1] *CMUcam* [online]. 2007 [cit. 2011-04-30]. Documentation – CMUcam. Dostupné z WWW: [<http://cmucam.org/wiki/Documentation>].
- [2] *Herodes.feld.cvut.cz* [online]. 2009 [cit. 2011-05-04]. Metoda nejmensich ctvercu. Dostupné z WWW: <<http://herodes.feld.cvut.cz/mereni/mnc/mnc.php>>.
- [3] NGUYEN, Viet, et al. *A Comparison of Line Extraction Algorithms using 2D Laser Rangefinder for Indoor Mobile Robotics*. Edmonton, Canada : IROS, 2005. 6 s.
- [4] *Arm-TX series 40 family: Instruction Manual: Version 6*. Faverges: Stäubli Faverges SCA, 2007. 84 s.
- [5] *VAL3 Reference Manual: Version 6*. Faverges: Stäubli Faverges SCA, 2008. 186 s.