

# **Fúze obrazových dat dvou kamer pracujících na různých vlnových délkách**

Fusion of image data from two video-cameras working  
with different wavelength

Bc. Jana Návratová

---

Diplomová práce  
2008



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
Ústav elektrotechniky a měření  
akademický rok: 2007/2008

## **ZADÁNÍ DIPLOMOVÉ PRÁCE**

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jana NÁVRATOVÁ**  
Studijní program: **N 3902 Inženýrská informatika**  
Studijní obor: **Bezpečnostní technologie, systémy a management**

Téma práce: **Fúze obrazových dat dvou kamer pracujících na různých vlnových délkách**

Zásady pro vypracování:

1. Nastudujte algoritmy, které jsou schopné řešit sesouhlasení obrazů získaných kamerami ve viditelné a infračervené oblasti spektra.
2. Rozhodněte se pro jeden algoritmus a ten zrealizujte ve formě programu.
3. Program otestujte na obrazových předlohách a výsledky uveďte ve své práci.

Rozsah práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **GONZALES, Rafael C. Digital image processing using MATLAB. [s.l.] : Upper Saddle River: Pearson/Prentice Hall, 2004. 2 sv. (xiv, 609 s.) ISBN 0-13-008519-7**
2. **JAIN, Anil K. Fundamentals of digital image processing. [s.l.] : Upper Saddle River : Prentice Hall, 1989. 2 sv. (xxi, 569 s.). ISBN 0-13-336165-9**
3. **PERŮTKA, Karel. MATLAB – Základy pro studenty automatizace a informačních technologií. [s.l.] : UTB ve Zlíně, 2005. 303 s. ISBN 80-7318-355-2**
4. **ZAPLATÍLEK, Karel, DOŇAR, Bohuslav. MATLAB – tvorba uživatelských aplikací. Praha : Nakladatelství BEN, 2004. 216 s. ISBN 80-7300-133-0**

Vedoucí diplomové práce:

**doc. RNDr. Vojtěch Křesálek, CSc.**

Ústav elektrotechniky a měření

Datum zadání diplomové práce:

**22. února 2008**

Termín odevzdání diplomové práce:

**4. června 2008**

Ve Zlíně dne 22. února 2008



prof. Ing. Vladimír Vašek, CSc.

*děkan*



doc. RNDr. Vojtěch Křesálek, CSc.

*ředitel ústavu*

## **ABSTRAKT**

Diplomová práce pojednává o fúzi obrazových dat ze dvou kamer. Pro řešení daného problému byl vytvořen program v prostředí MATLAB 7.0 se dvěma algoritmy.

První vyžaduje přímou interakcí s uživatelem, kde si uživatel vybere v prvním i druhém obrazu dva body, podle nichž se složí výsledný obraz.

Druhý algoritmus pracuje na principu diskrétní konvoluce a korelačních koeficientů, kde uživatelské vstupy algoritmu jsou maximální a minimální úhel rotace, krok rotace, maximální a minimální změna měřítka a krok změny měřítka.

Klíčová slova: MATLAB, digitální zpracování obrazu, diskrétní konvoluce, korelační koeficient, hranový detektor

## **ABSTRACT**

This Master thesis deals with fusion of image data from two video-cameras working with different wavelength. A program with two algorithms was developed in MATLAB 7.0 environment as a solution of this problem.

The first algorithm requires a direct interaction with a user. The user chooses two points in two input images. After it these images will be combined.

The second algorithm works with discrete convolution and correlation coefficient. The user chooses input parameters (maximum and minimum angle of rotation, step of rotation, maximum and minimum size of scale, step of scale) for fusion of image data.

Keywords: MATLAB, digital image processing, discrete convolution, correlation coefficient, edge detector

Ráda bych touto cestou poděkovala vedoucímu diplomové práce panu doc. RNDr. Vojtěchu Křesálkovi, CSc. za odborné vedení, rady a připomínky, které mi při řešení mé práce poskytoval.

Samozřejmě děkuji svým rodičům za jejich nezištnou oporu po celou dobu mých studií a svému příteli, který se mi snažil vytvářet ideální prostředí pro tvorbu mé práce.

V neposlední řadě chci poděkovat panu Ing. Janu Machylovi z VTÚVM Slavičín za poskytnutí testovacích obrazů.

*Motto*

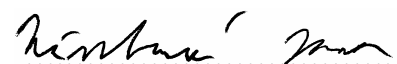
„ Science without religion is lame,  
religion without science is blind. “

„ Věda bez víry je chromá,  
víra bez vědy je slepá. “

ALBERT EINSTEIN (\*1879 – †1955)

Prohlašuji, že jsem na diplomové práci pracovala samostatně a použitou literaturu jsem citovala. V případě publikace výsledků, je-li to uvolněno na základě licenční smlouvy, budu uvedena jako spoluautorka.

Ve Zlíně 31.5.2008



Podpis diplomanta

**OBSAH**

<b>ÚVOD</b> .....	<b>7</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>9</b>
<b>1 ELEKTROMAGNETICKÉ SPEKTRUM</b> .....	<b>10</b>
1.1 VIDITELNÁ OBLAST SPEKTRA.....	10
1.2 INFRAČERVENÁ OBLAST SPEKTRA.....	11
<b>2 MATLAB</b> .....	<b>12</b>
2.1 PREZENTACE OBRAZŮ V MATLABU.....	13
2.2 DETEKCE HRAN .....	14
2.2.1 Hranový detektor Canny.....	14
<b>3 MATEMATICKÝ ZÁKLAD</b> .....	<b>16</b>
3.1 DISKRÉTNÍ KONVOLUCE .....	16
3.2 KORELAČNÍ KOEFICIENTY .....	16
3.3 ROTACE.....	17
3.4 ZMĚNA MĚŘÍTKA .....	18
<b>II PRAKTICKÁ ČÁST</b> .....	<b>19</b>
<b>4 PROGRAM SESOUHLASENÍ OBRAZŮ</b> .....	<b>20</b>
4.1 SPUŠTĚNÍ PROGRAMU A NAČTENÍ OBRAZŮ.....	21
4.2 DVA BODY.....	22
4.3 TRANSFORMACE.....	24
4.4 ZOBRAZENÍ A ULOŽENÍ VÝSLEDKU .....	26
4.5 UKONČENÍ PROGRAMU .....	27
<b>5 POPIS ALGORITMŮ PROGRAMU</b> .....	<b>28</b>
5.1 DVA BODY.....	28
5.2 TRANSFORMACE.....	29
5.3 SROVNÁNÍ ALGORITMŮ .....	33
<b>6 VÝSLEDKY TESTOVÁNÍ</b> .....	<b>34</b>
<b>ZÁVĚR</b> .....	<b>37</b>
<b>CONCLUSION</b> .....	<b>38</b>
<b>SEZNAM POUŽITÉ LITERATURY</b> .....	<b>39</b>
<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK</b> .....	<b>41</b>
<b>SEZNAM OBRÁZKŮ</b> .....	<b>42</b>
<b>SEZNAM PŘÍLOH</b> .....	<b>44</b>

## ÚVOD

Již přes dvě stě let zná lidstvo infračervené záření. Používá se v mnoha oblastech lidského působení. V bezpečnostních systémech se s ním lze setkat hlavně u detektorů pohybu, infračervených závor. Další z možností použití je také snímání střeženého prostoru infračervenou kamerou.

V dnešní době je možno se setkat s aplikací dvou kamer o různých vlnových délkách zabudovaných v jediném bezpečnostním systému. Jedná se zejména o použití ve vojenské sféře. Na obrázku (Obr. 1) je možno vidět průzkumný a pozorovací systém POS [1], který se používá pro střežení pozemku. Obsahuje kamery, které pracují ve viditelné a infračervené oblasti spektra. Sesouhlasením obrazů získaných z těchto kamer se dosáhne lepšího přehledu o hlídaném prostoru, než za použití jen jednoho typu kamery.

Sloučení obrazů může sloužit pro ověření objektů nacházejících se v snímaném teritoriu. Po nasnímání teritoria kamerou pracující ve viditelné oblasti spektra, je uživatel schopen v noci během použití infračervené kamery zjistit sloučením obrazů, zda se v teritoriu nenachází subjekt, který tam nepatří.

Cílem práce je navrhnout algoritmy řešící fúzi obrazových dat ze dvou kamer pracujících na různých vlnových délkách.

V teoretické části diplomové práce budou popsány důležité rovnice, ze kterých se bude později vycházet při tvorbě algoritmů a vysvětlen princip předzpracování obrazů.

V praktické části bude pro odzkoušení těchto algoritmů vytvořen program v prostředí MATLAB. Pomocí obrazů poskytnutých od pana Ing. Jana Machyla z VTÚVM Slavičín se provede testování správného fungování programu.

V závěru práce se zhodnotí výsledky testování. Testovací obrazy z infračervené kamery budou pro větší přehlednost zbaveny (ořezány) černých okrajů.



*Obr. 1. Lehký průzkumný a pozorovací systém POS [1]*



## **I. TEORETICKÁ ČÁST**

# 1 ELEKTROMAGNETICKÉ SPEKTRUM

Elektromagnetickým zářením se rozumí příčné vlnění magnetického a elektrického pole. Závislost elektromagnetického záření o vlnové délce  $\lambda$  na frekvenci  $f$  znázorňuje následující vztah (Rovnice 1):

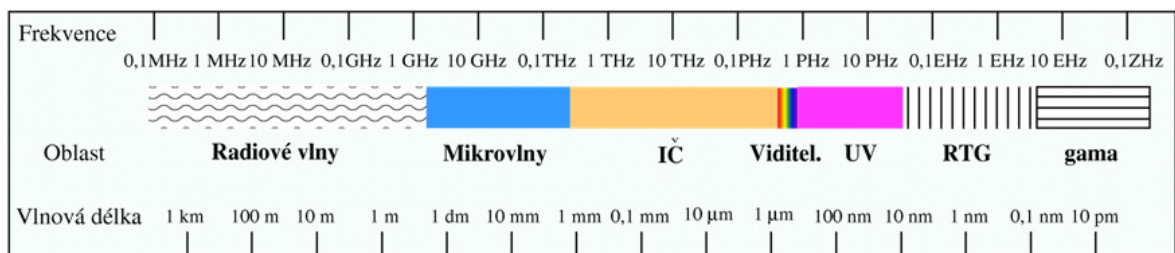
$$\lambda = \frac{c}{f} \quad (1)$$

kde:  $\lambda$  – vlnová délka ve vakuu [m]

$f$  – frekvence kmitavého pohybu [Hz]

$c$  – rychlost šíření elektromagnetických vln (ve vakuu rovno 299 792 458 m/s)

Elektromagnetická záření rozdílných vlnových délek vytváří spektrum elektromagnetického záření (rovněž zvané Maxwellova duha). Elektromagnetické spektrum se dělí pomocí vlnové délky na několik druhů (Obr. 2). Druh vlnění se určuje také dle vzniku. Díky tomu dochází k prolínání mezi jednotlivými druhy elektromagnetického spektra. [2]



Obr. 2. Elektromagnetické spektrum [2]

## 1.1 Viditelná oblast spektra

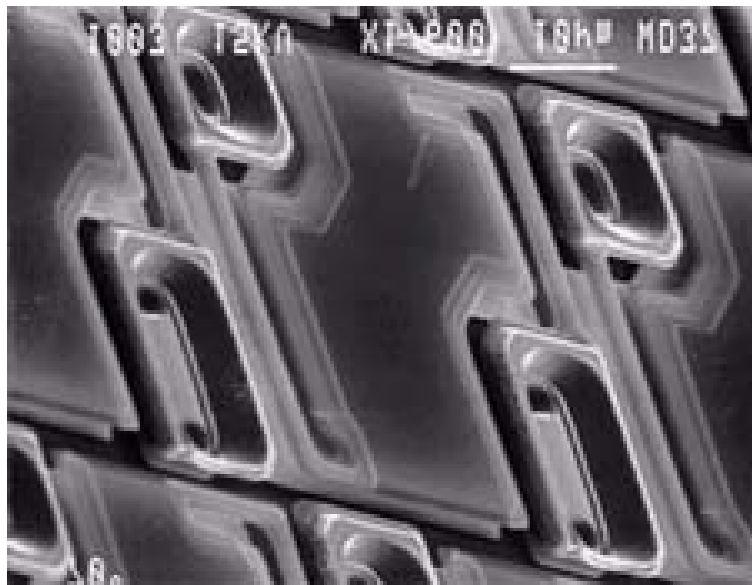
Lidské oko je citlivé na viditelnou oblast spektra, také nazývanou viditelné světlo. Jedná se o oblast spektra, která se nalézá mezi ultrafialovou oblastí spektra a infračervenou oblastí spektra, tedy mezi vlnovou délkou 0,35 μm až 0,75 μm. [3]

Denní kamery snímají viditelnou oblast spektra a infračervenou složku záření odfiltrují. Kamery s vyšší spektrální citlivostí se používají jako soumrakové kamery, jenž snímají prostor i za zhoršených světelných podmínek. Soumrakové kamery pracují jak s viditelnou tak i s infračervenou oblastí spektra. Toho je docíleno odejmutím infračerveného filtru.

## 1.2 Infračervená oblast spektra

Infračervená oblast spektra (též IR) pokrývá vlnová délka od 0,75  $\mu\text{m}$  do 1000  $\mu\text{m}$ . Je patrné, že IR má daleko větší rozsah než viditelné světlo. [3]

Infračervené kamery (termovize) pracují s infračervenou oblastí spektra. Narozdíl od kamer, které snímají viditelnou oblast spektra se obvykle používají místo CCD čipů mikrobolometry (Obr. 3). Výhodou oproti denních kamer je používání v noci bez přisvětlení a také možnost odhalení osoby, která se schovává v těžko přehledných místech. [4]



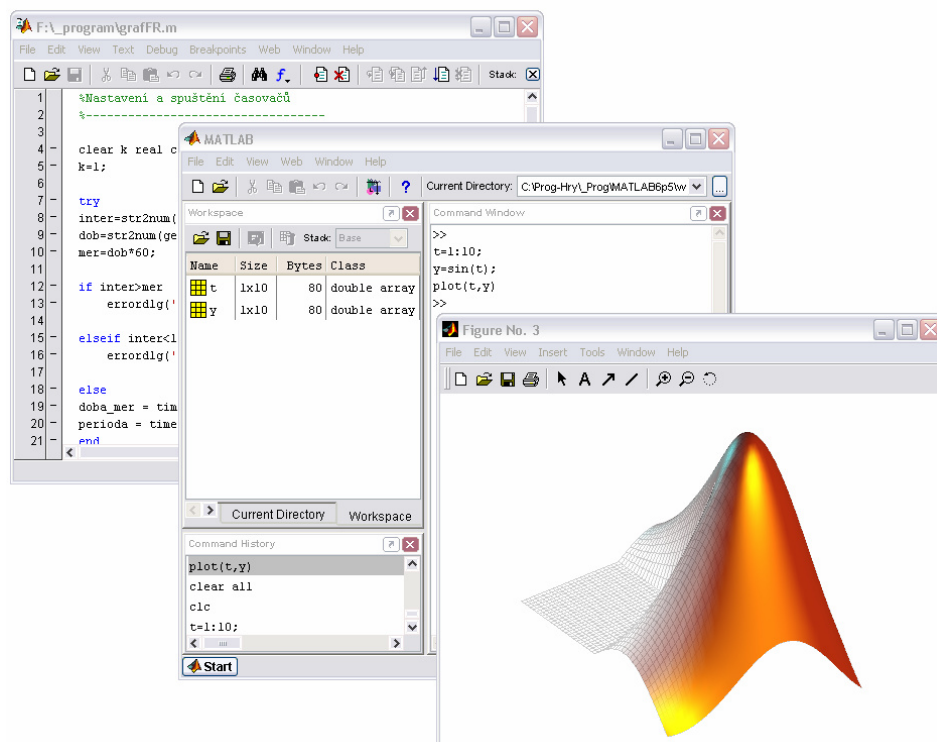
*Obr. 3. Detail jedné plošky mikrobolometru [4]*

## 2 MATLAB

„MATLAB<sup>®</sup> (Obr. 4) je špičkové integrované prostředí pro vědeckotechnické výpočty, modelování, návrhy algoritmů, simulace, analýzu a prezentaci dat, měření a zpracování signálů, návrhy řídicích a komunikačních systémů.

Otevřená architektura MATLABu vedla ke vzniku knihoven funkcí, nazývaných toolboxy, které rozšiřují použití programu v příslušných vědních a technických oborech.“ [5]

„*Image Processing Toolbox* je výkonný, pružný a snadno ovladatelný nástroj pro zpracování a analýzu obrazu. Na základě mohutného výpočetního potenciálu MATLABu jsou vybudovány nadstavby pro návrhy filtrů, rekonstrukci a analýzu obrazů, dále nadstavby pro manipulaci s barvami, geometrií a strukturou obrazů včetně 2-D transformací.“ [6]

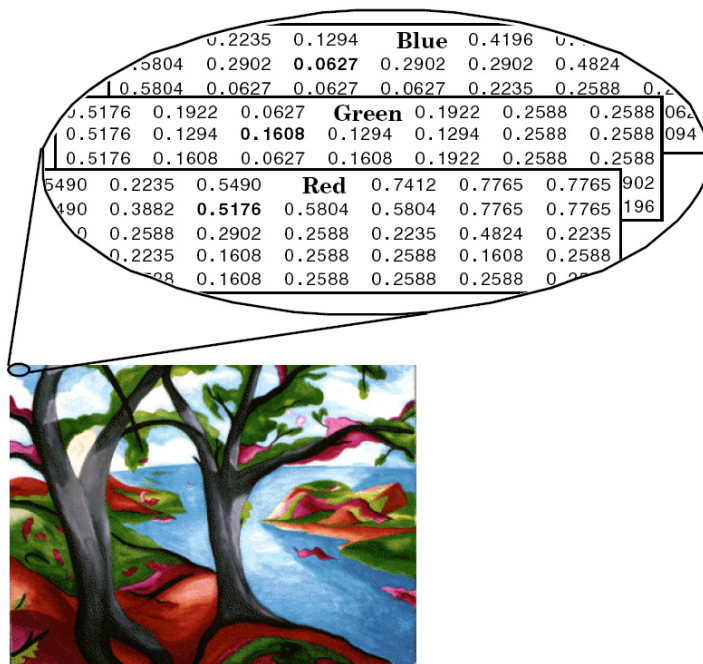


Obr. 4. Prostedí MATLAB

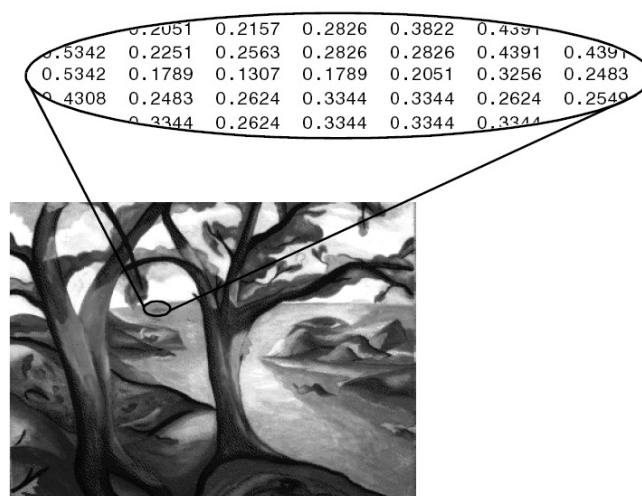
Pro bližší seznámení s programovacím prostředím MATLAB slouží knihy od autorů Zaplatílek Karel a Doňar Bohuslav [7] a [8].

## 2.1 Prezentace obrazů v MATLABu

Obrazy z kamer pracujících na různých vlnových délkách jsou v MATLABu interpretovány jako matice o rozměrech  $x$ ,  $y$ ,  $z$ . Kde  $x$  je počet sloupců odpovídající šířce obrazu,  $y$  je počet řádků odpovídající výšce obrazu. Rozměr  $z$  je roven počtu složek barev v obrazu. V případě barevného obrazu je tato hodnota tři, jelikož jsou použity složky RGB (Obr. 5). Při použití stupnice šedi je tato hodnota jedna (Obr. 6) a uživatel si může nastavit interval, ve kterém bude barva nabývat hodnotu.



Obr. 5. Obraz a matice RGB [9]



Obr. 6. Obraz a matice stupnice šedi [9]

## 2.2 Detekce hran

Pomocí předzpracování obrazů, lze dosáhnout lepších výsledků v následné manipulaci s obrazem. Jednou z možností je použití detekce hran v obraze. K tomu slouží v prostředí MATLAB funkce **edge**:

```
Výstupní obraz = edge(vstupní obraz, hranový detektor, práh citlivosti)
```

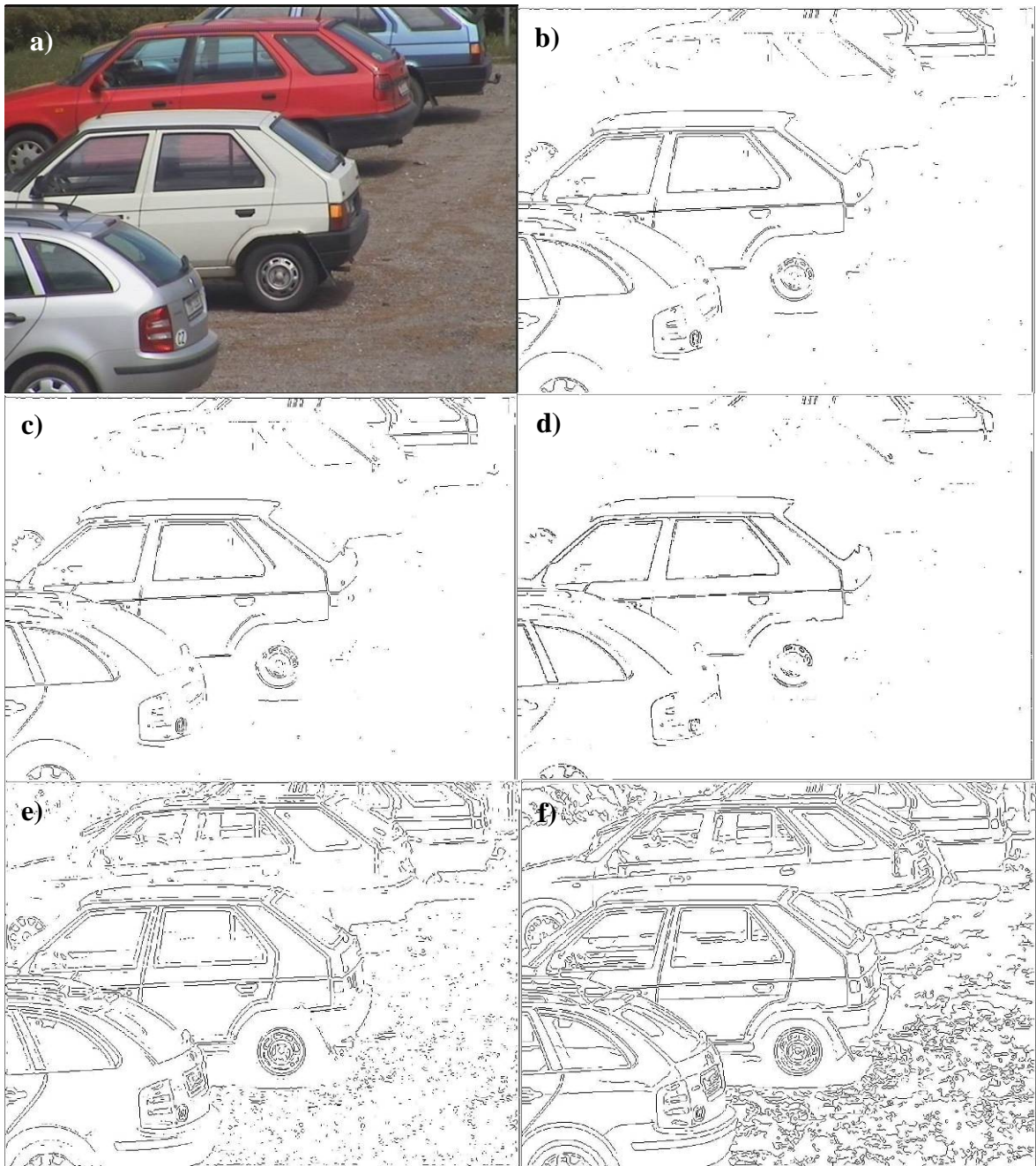
Vstupní obraz je v stupnici šedi a výstupní obraz nabývá binárních hodnot (je černobílý) a má stejný rozměr jako originální obraz. Bílá barva (hodnota jedna) zobrazuje nalezené hran y a černou barvu (hodnota nula) mají ostatní pixely. Při porovnávání hranových detektorů byla nastavena automatická volba parametru `práh citlivosti`. MATLAB používá několik základních typů hranových detektorů (Obr. 7 - pro větší přehlednost byly černobílé obrazy invertovány) [10]:

- Hranové detektory *Sobel* (Obr. 7b) a *Prewitt* (Obr. 7c) dosahují podobných výsledků. Odlišují se jen nepatrně v méně zřetelných hranách.
- Hranový detektor *Roberts* (Obr. 7d) se řadí mezi starší a jednodušší hranové detektory. Nalezl nejmenší počet hran, čímž se řadí k nejméně přesným metodám hledání hran.
- Velice přesné jsou hranové detektory *Laplacian of Gaussian* (zkráceně LoG) a *Canny*. U hranového detektoru *LoG* (Obr. 7e) se projevuje chybná detekce hrany u okraje obrazu. Hranový detektor *Canny* (Obr. 7f) se jeví jako nejlepší volba pro další práci s obrazem.

### 2.2.1 Hranový detektor Canny

Jedná se o algoritmus, který lze shrnout do následujícího postupu [11]:

1. Redukce šumu použitím Gaussova filtru.
2. Určení gradientů pomocí metody blízkého bodu, Sobel nebo Prewitt.
3. Nalezení lokálních maxim gradientů.
4. Odstranění nevýznamných hran pomocí nastavení dvou prahů.



Obr. 7. a) originální obraz; b) Sobel; c) Prewitt; d) Roberts; e) LoG; f) Canny

### 3 MATEMATICKÝ ZÁKLAD

Přehled matematiky použité v algoritmech.

#### 3.1 Diskrétní konvoluce

Pro řešení sesouhlasení obrazů, byla zvolena funkce diskrétní konvoluce (Rovnice 2). [12]

$$C_{a,b} = \sum_{x=-k}^k \sum_{y=-l}^l f_{a-x,b-y} \cdot h_{a,b} \quad (2)$$

kde:  $C_{a,b}$  – výstupní obraz diskrétní konvoluce

$f_{a-x,b-y}$  – vstupní statický obraz

$h_{a,b}$  – vstupní dynamický obraz o rozměrech  $\langle -k,k \rangle$  a  $\langle -l,l \rangle$

$a, b, x, y$  – souřadnice matice obrazů

Výstupní obraz diskrétní konvoluce  $C_{a,b}$  vznikne posouváním vstupního dynamického obrazu  $h_{a,b}$  po vstupním statickém obraze  $f_{a-x,b-y}$  a vypočítáním jejich součtu na dané pozici. Rozměr výstupního obrazu  $C_{a,b}$  lze získat součtem rozměru vstupních obrazů mínus jedna, jak v ose x tak i v ose y.

#### 3.2 Korelační koeficienty

Výpočtem korelačních koeficientů (Rovnice 3) se získá matice, která nabývá hodnot v intervalu  $\langle -1,1 \rangle$ . Obecně platí, že čím je větší absolutní hodnota korelačního koeficientu, tím větší je shoda mezi statickým a dynamickým obrazem. [13]

$$\rho_{i,j} = \frac{C_{a,b}}{\sqrt{\sigma_f^2 \cdot \sigma_h^2}} \quad (3)$$

kde:  $\rho_{a,b}$  – korelační koeficient v hodnotě  $\langle -1,1 \rangle$

$\sigma_f^2$  – rozptyl statického obrazu

$\sigma_h^2$  – rozptyl dynamického obrazu



### 3.3 Rotace

Pro otočení obrazu o zvolený úhel  $\alpha$  slouží transformační matice (Rovnice 4). [12]

$$R(\alpha) = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4)$$

V MATLABu je tato matice zastoupená funkcí **imrotate**:

Výstupní obraz = `imrotate(vstupní obraz,úhel,metoda interpolace,ořez)`  
 Parametr `metoda interpolace` nabývá hodnot: *nearest*, *bilinear* a *bicubic*. Nejpřesněji se jeví použití bikubické interpolace. Další parametr `ořez` slouží pro ořezání výstupního obrazu na stejnou velikost jakou má obraz originální (Obr. 8). Přesahující okraje jsou automaticky vyplněny černou barvou.



Obr. 8. Vlevo-původní velikost obrazu; vpravo-výsledný obraz po ořezání

Pokud se má obraz otáčet kolem předem zvoleného počátku, musí se přepočítat jeho souřadnice  $X$  a  $Y$  (Rovnice 5) [12].

$$\begin{aligned} X' &= X \cos \alpha - Y \sin \alpha \\ Y' &= X \sin \alpha + Y \cos \alpha \end{aligned} \quad (5)$$

kde:  $X, Y$  – souřadnice zvoleného počátku

$\alpha$  – úhel rotace

$X', Y'$  – výsledné souřadnice otočeného bodu

### 3.4 Změna měřítka

Zvětšení, popřípadě zmenšení obrazu se provádí pomocí následující transformační matice (Rovnice 6), kde  $s_x$  a  $s_y$  jsou koeficienty změny měřítka na ose x a ose y. V případě zachování poměru stran se tyto koeficienty musejí rovnat. [12]

$$S(s_x, s_y) = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (6)$$

Funkce pro změnu měřítka v MATLABu je **imresize**:

```
Výstupní obraz = imresize(vstupní obraz, změna, metoda interpolace)
```

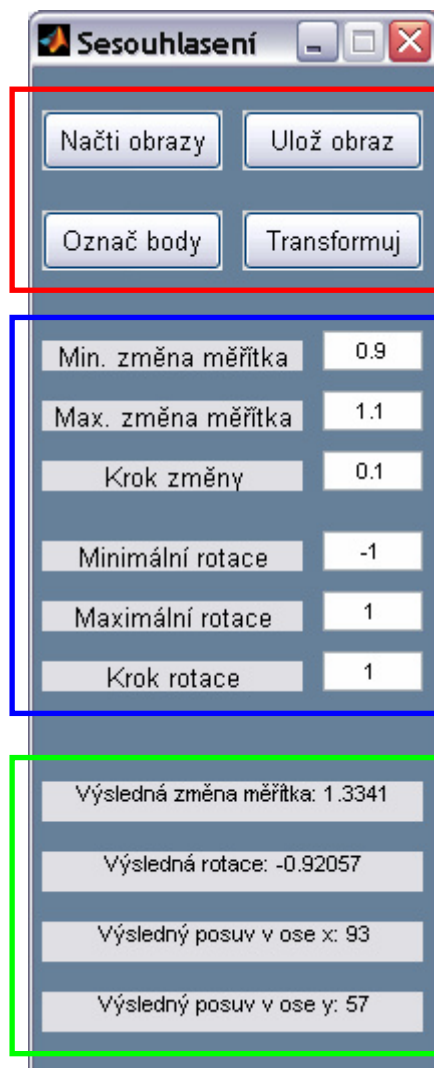
Funkce **imresize** má obdobné parametry jako funkce **imrotate**. I zde se jeví jako nejvhodnější interpolace bikubická. Parametr *změna* udává, zda se jedná o zvětšení nebo zmenšení obrazu. Pokud je *změna* v rozsahu (0, 1), jedná se o zmenšení obrazu. Rozsah (1, +∞) slouží pro zvětšení obrazu. Po dosažení hodnoty jedna zůstává výsledný obraz stejný jako vstupní obraz.

## **II. PRAKTICKÁ ČÁST**

## 4 PROGRAM SESOUHLASENÍ OBRAZŮ

Pro řešení sesouhlasení obrazů získaných z různých kamer, pracujících ve viditelné a infračervené oblasti spektra, bylo zvoleno vývojové prostředí MATLAB, ve kterém byl vytvořen program **Sesouhlasení obrazů** (Obr. 9). Program má intuitivní ovládání. Skládá se ze tří oblastí:

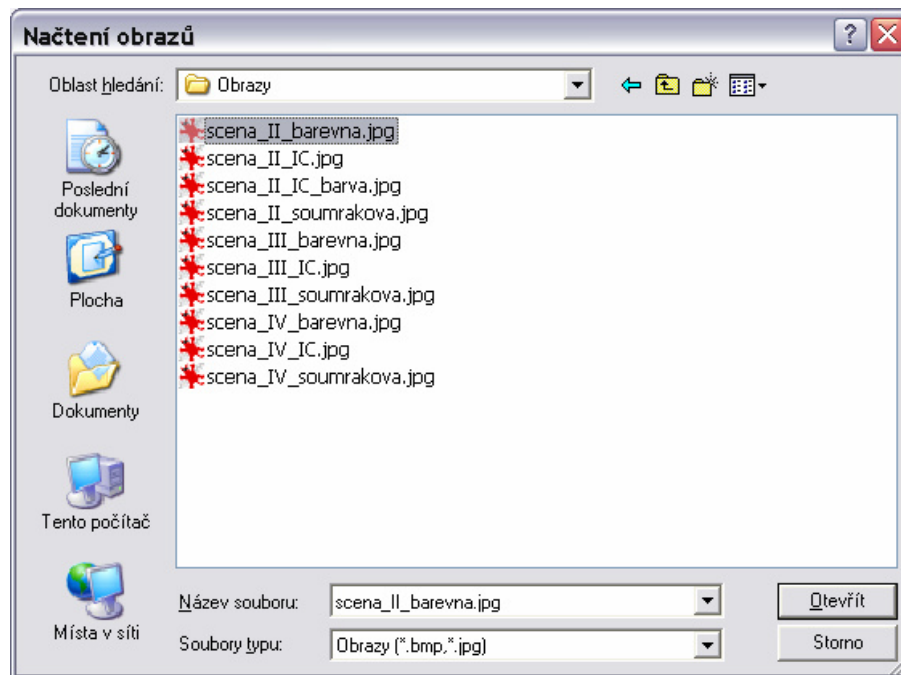
- První je oblast tlačítek (označena červeně), které slouží k načtení a uložení obrazů a spuštění jednoho z algoritmu řešení sesouhlasení obrazů.
- Druhá oblast (označena modře) se skládá z šesti přepisovatelných textových polí, které slouží k přesnému nastavení transformace obrazů.
- V poslední oblasti (označena zeleně) se zobrazují údaje o výsledném obrazu.



Obr. 9. Sesouhlasení obrazů

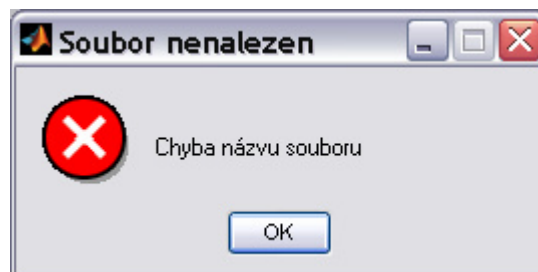
## 4.1 Spuštění programu a načtení obrazů

Program se spouští souborem *sesouhlaseni.exe*, popřípadě v prostředí MATLAB, spuštěním funkce **sesouhlaseni**. V pravém horním rohu monitoru se zobrazí hlavní okno programu (Obr. 9). Po stisknutí tlačítka *Načti obrázky* se zobrazí dialogové okno (Obr. 10) pro výběr dvou obrazů ve formátu *\*.jpg* popřípadě *\*.bmp*. Uživatel si vybere obrázky, které chce sloučit. Ty se následně vykreslí do vlastních oken zhruba uprostřed monitoru.



Obr. 10. Načtení obrazů

Pokud uživatel zadá chybnou cestu k načtení obrazů, zobrazí se příslušné dialogové okno (Obr. 11), které uživatele na tuto skutečnost upozorní.



Obr. 11. Varovné okno pro načítání

Dalším krokem je zvolení jednoho z algoritmu řešení sesouhlasení vybraných obrazů. Nyní následuje podrobný popis jednotlivých algoritmů.

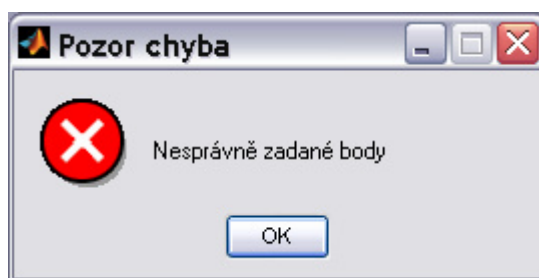
## 4.2 Dva body

Pro první algoritmus “Dva body“ slouží tlačítko *Označ body*. Po stisknutí si uživatel zvolí dva body, které si navzájem odpovídají na obou obrazech. Body se vybírají levým tlačítkem myši a výběr se ukončuje stisknutím pravého tlačítka, popřípadě dvojklikem. Pokud chce uživatel smazat vybrané body, slouží k tomu tlačítka na klávesnici *Delete*, popřípadě *Backspace*. Vybrané body se označují na obrazech hvězdou (Obr. 12).



Obr. 12. Označení vybraných bodů hvězdou

Jestliže uživatel chybně vybere body na obrazech, zobrazí se dialog s upozorněním na tuto chybu (Obr. 13).



Obr. 13. Nesprávně zadané body

Výstupem sesouhlasení obrazu z viditelné oblasti spektra (Obr. 14) a obrazu z infračervené oblasti spektra (Obr. 15) je obraz získaný alfa mícháním (Obr. 16). (Pro větší přehlednost jsou vybrané body na vstupních obrazech zvýrazněny žlutým kruhem.) Výsledný obraz se vykreslí do středu monitoru.



Obr. 14. Obraz z kamery ve viditelné oblasti spektra



Obr. 15. Obraz z kamery v infračervené oblasti spektra



Obr. 16. Výsledný sesouhlasený obraz algoritmu Dva body

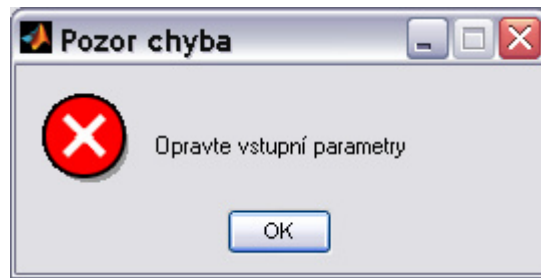
### 4.3 Transformace

Druhým algoritmem řešení je "Transformace". Nejprve si uživatel nastaví parametry:

- minimální změna měřítka – nejmenší doporučená hodnota je 0.5
- maximální změna měřítka – čím větší hodnota, tím větší doba výpočtu
- krok měřítka
- minimální změna rotace
- maximální změna rotace
- krok rotace

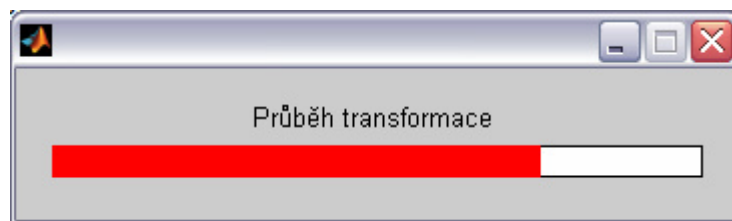
Parametry vymezí oblast, ve které se bude hledat sesouhlasení obrazů. Program je ošetřen před nesprávným zadáním parametrů (Obr. 17).





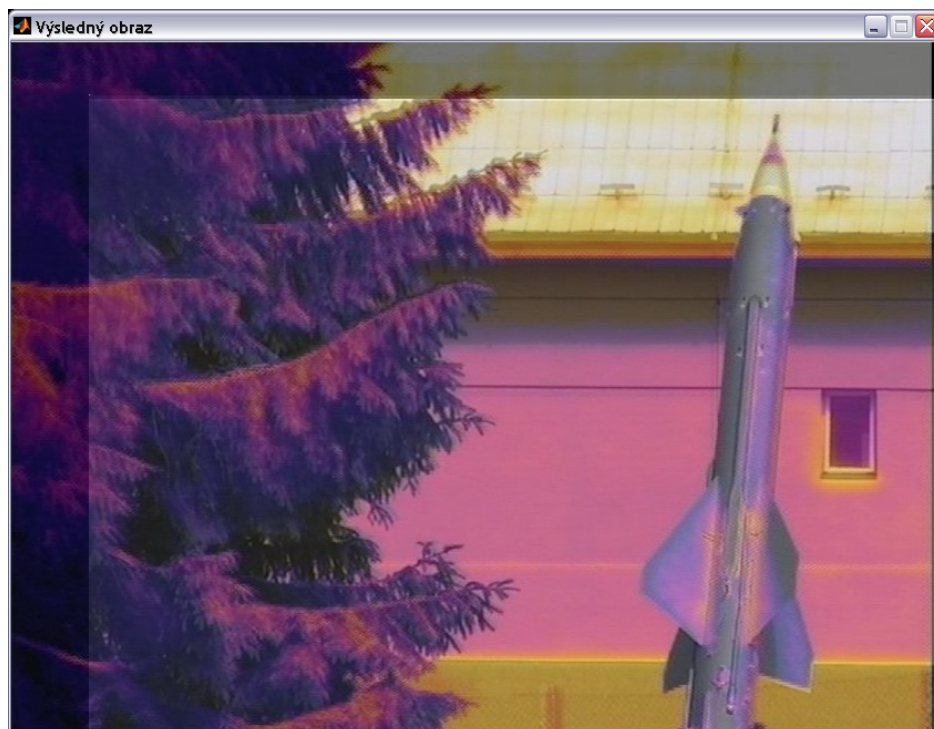
*Obr. 17. Opravte vstupní parametry*

Po stisknutí tlačítka *Transformuj* se zobrazí dialogové okno (Obr. 18), které znázorňuje časový průběh výpočtu.



*Obr. 18. Zobrazení průběhu výpočtu*

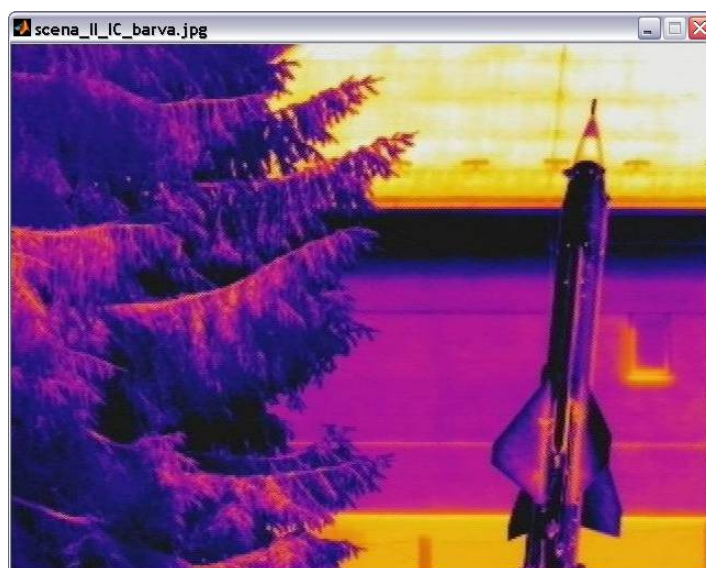
Výsledný obraz (Obr. 19) vznikne sloučením obrazu z viditelné oblasti spektra (Obr. 20) a obrazu z infračervené oblasti spektra (Obr. 21). Výsledný obraz je vykreslen alfa mícháním do vlastního okna uprostřed monitoru.



*Obr. 19. Výsledný sesouhlasený obraz algoritmu Transformace*



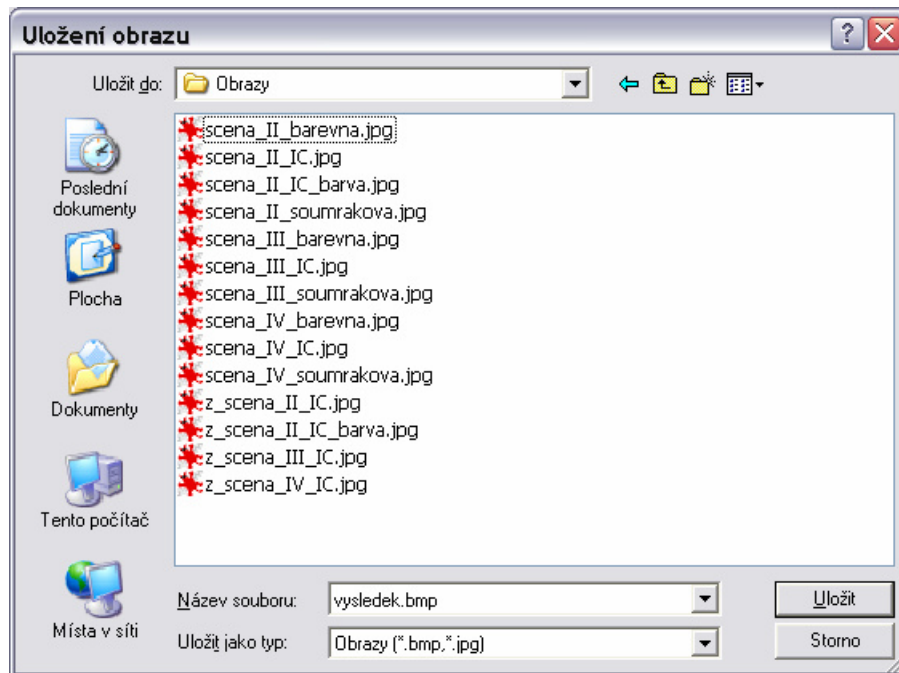
Obr. 20. Obraz z kamery ve viditelné oblasti spektra



Obr. 21. Obraz z kamery v infračervené oblasti spektra

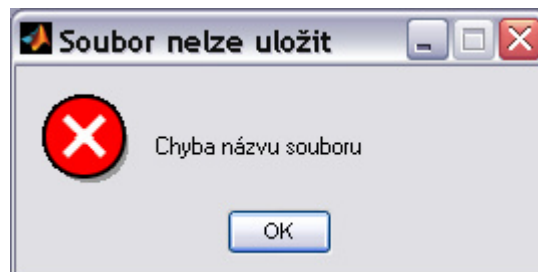
#### 4.4 Zobrazení a uložení výsledku

Výsledná rotace, změna měřítka a posuv v ose x a y se vypíše do dolní části hlavního okna. Úspěšným provedením jednoho z algoritmu řešení, se zpřístupní tlačítko *Ulož obraz*, které slouží pro uložení výsledného obrazu. Po použití tohoto tlačítka se objeví dialogové okno (Obr. 22) pro uložení obrazu na disk ve formátu *\*.bmp* nebo *\*.jpg*. Předdefinovaný název souboru je *vysledek.bmp*.



Obr. 22. Uložení obrazu

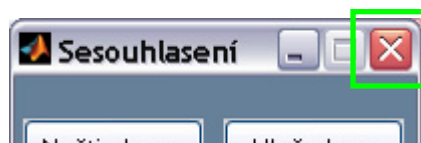
Pokud uživatel zadá chybně cestu pro uložení obrazu, zobrazí se dialogové okno (Obr. 23) s upozorněním na tuto skutečnost.



Obr. 23. Varovné okno pro ukládání

## 4.5 Ukončení programu

Program se zavírá kliknutím na červený křížek (Obr. 24), který se nachází vpravo nahoře v hlavním okně programu.



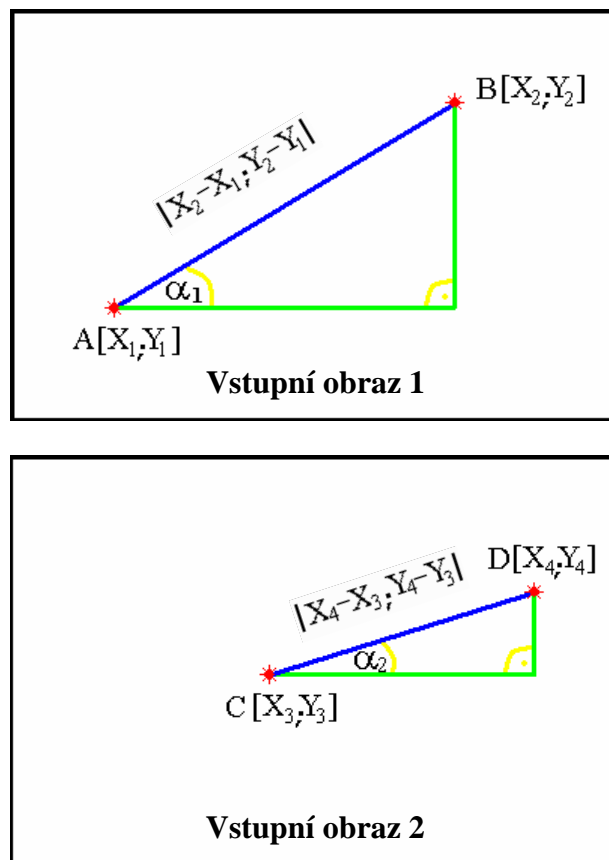
Obr. 24. Ukončení programu

## 5 POPIS ALGORITMŮ PROGRAMU

V programu jsou implementovány dva algoritmy sesouhlasení obrazů. Jedná se o algoritmy “Dva body“ a “Transformace“.

### 5.1 Dva body

Na začátku algoritmu se zjistí souřadnice uživatelem zvolených bodů A, B, C a D ve vstupních obrazech (Obr. 25). Z těchto souřadnic se vypočítají úsečky mezi body a úhly  $\alpha_1$  a  $\alpha_2$ . Poměrem velikosti úseček se získá výsledná změna měřítka a rozdílem úhlů se zjistí velikost vzájemného otočení obrazů. Pro otočení obrazu kolem zvoleného středu se použije rovnice (Rovnice 5). Vzájemný posuv je dán vzdáleností úseček. Získané hodnoty se aplikují na jeden z vstupních obrazů pomocí funkcí pro rotaci **imrotate** a změny měřítka **imresize**. Sesouhlasení obrazů se provádí vytvořením nového obrazu, do kterého jsou umístěny oba vstupní obrazy.



Obr. 25. Schématický náčrt vstupních obrazů

## 5.2 Transformace

Algoritmus “Transformace“ využívá diskrétní konvoluce (Rovnice 2) a metody korelačních koeficientů (Rovnice 3). Uživatelem zvolené obrázky (Obr. 20 a Obr. 21) se převedou pomocí funkce **rgb2gray** do stupnice šedi (Obr. 26 a Obr. 27).



*Obr. 26. Obrázek z kamery ve viditelné oblasti spektra*



*Obr. 27. Obrázek z kamery v infračervené oblasti spektra*

Následně se vykoná funkce **edge**, která v obrazech za použití hranového detektoru *canny* vyhledá a označí hrany (Obr. 28 a Obr. 29 - pro větší přehlednost byly obrázky invertovány). Dalším krokem je porovnání velikosti zpracovaných vstupních obrazů. Z důvodů urychlení následujících výpočtů se dále pracuje s menším obrazem. V tomto případě je to obraz z kamery v infračervené oblasti spektra (Obr. 21).



*Obr. 28. Obraz z kamery ve viditelné oblasti spektra*



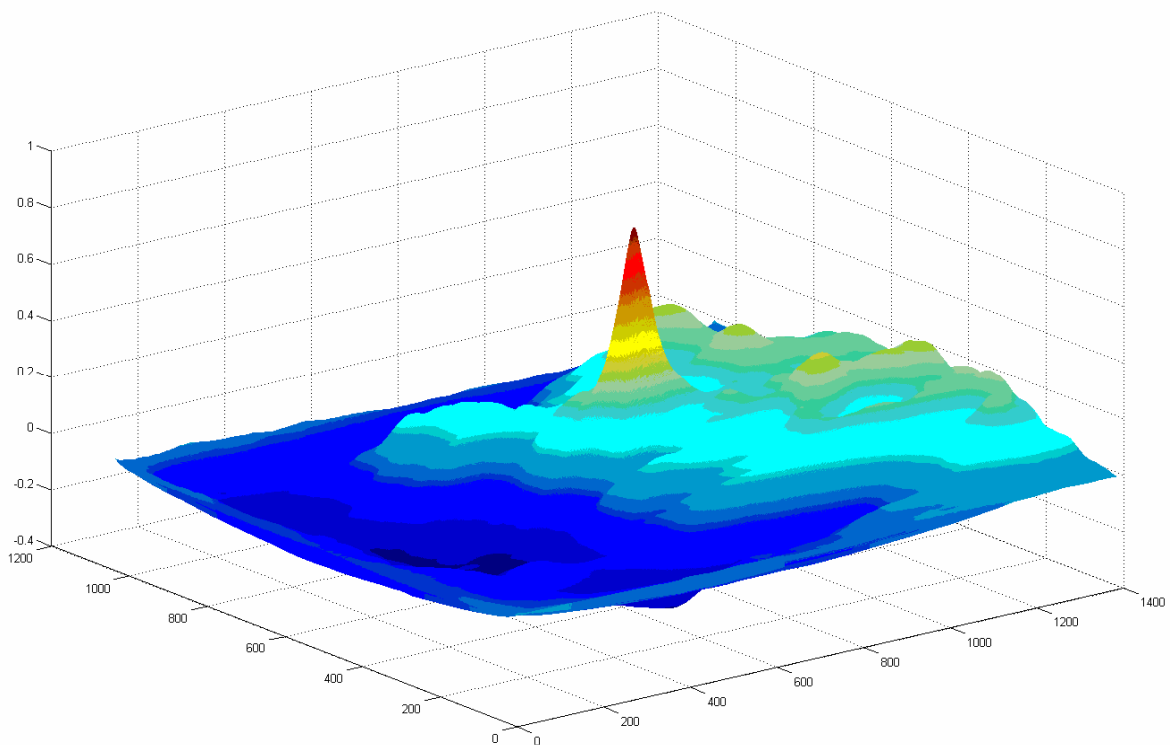
*Obr. 29. Obraz z kamery v infračervené oblasti spektra*

Posléze probíhají cykly otáčení a změny měřítka obrazu podle uživatelem zvolených parametrů (minimální a maximální změna měřítka, krok měřítka, minimální a maximální změna rotace, krok rotace). V každém cyklu se aplikuje rovnice diskrétní konvoluce (Rovnice 2) a rovnice pro výpočet korelačních koeficientů (Rovnice 3). V prostředí MATLAB zahrnuje obě zmíněné rovnice funkce **normxcorr2**:

Výstupní obraz = normxcorr2(vstupní obraz 2, vstupní obraz 1)

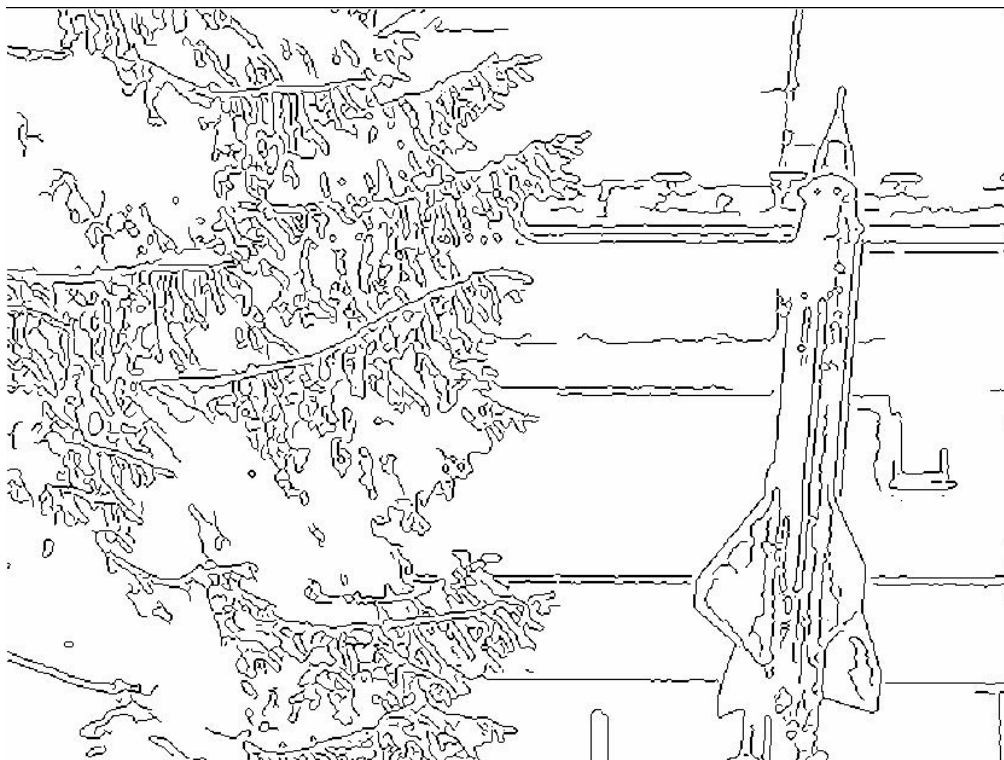
Podmínkou této funkce je, že vstupní obraz 1 musí být větší v obou rozměrech než vstupní obraz 2. Toho je docíleno doplnění vstupního obrazu 1 nulami (černá barva).

Příklad grafického znázornění funkce **normxcorr2** je vidět na obrázku (Obr. 30). Nalezením nejvyšší hodnoty z maxim korelačních matic ze všech cyklů lze zjistit hodnotu rotace, změny měřítka a posuvu zpracovávaného obrazu pro nejpřesnější sesouhlasení vstupních obrazů.



Obr. 30. Výsledek funkce **normxcorr2**

Na obrázcích (Obr. 31 - pro větší přehlednost byl obraz invertován a Obr. 32) je vidět obraz z kamery v infračervené oblasti spektra, který již byl potočen a v tomto případě zvětšen o zjištěnou hodnotu. Sloučením tohoto zpracovaného vstupního obrazu (Obr. 32) a druhého vstupního obrazu (Obr. 20) se získá výsledný obraz (Obr. 19), který se vykreslí do nového okna.



*Obr. 31. Obraz z kamery v infračervené oblasti spektra*



*Obr. 32. Obraz z kamery v infračervené oblasti spektra*



### 5.3 Srovnání algoritmů

Oba algoritmy mají své výhody a nevýhody.

Algoritmus “Dva body“ je výpočetně rychlejší než algoritmus “Transformace“, protože u něho příliš nezáleží na velikosti zpracovávaných obrazů. Naopak nevýhodou je menší přesnost výsledného sesouhlasení, která je daná zvolením bodů uživatelem. Je taky náchylnější na chyby, kterých se může dopustit uživatel například špatným pořadím vybraných bodů.

Druhý algoritmus “Transformace“ se může urychlit zvolením menšího rozsahu rotace a změny měřítka. Také pokud se zvětší krok rotace a krok změny měřítka, vede to k urychlení výpočtu, protože proběhne méně cyklů. Standardně je vše nastaveno na co nejmenší počet cyklů. I zde však může dojít k chybě na straně uživatele zadáním rozsahu parametrů, ve kterém se výsledek nenachází.

## 6 VÝSLEDKY TESTOVÁNÍ

Celkem byly otestovány tři různé scény, které zahrnovaly obrazy z kamer pracujících ve viditelné a infračervené oblasti spektra (například Obr. 33 a Obr. 34). Po testování těchto obrazů se dospělo k závěru že:

- u algoritmu “Dva body“ je výsledek závislý na velikosti úhlu otočení a změny měřítka obrazu. Čím jsou tyto hodnoty vyšší, tím více záleží na správnosti zvolených bodů uživatelem. U testovaných obrazů se úhel otočení pohyboval kolem dvou stupňů.
- u algoritmu “Transformace“ se jako problematické jeví použití detekce hran v případě, že vyhledávací cyklus projde přes úhel rotace nula. Zde dochází k nalezení maxima, i přestože skutečná hodnota rotace je jiná. Je to zapříčiněno výraznějšími liniemi hran u nerotovaného obrazu.
- u algoritmu “Transformace“ bez použití detekce hran nedochází k problému s nulovou hodnotou u rotace. Nevýhodou však je, že nemá tak přesné výsledky jako za použití detekce hran.

Jako příklad možných chybných výsledků získaných sloučením obrazů (Obr. 33 a Obr. 34) slouží výstupní obrazy (Obr. 35 a Obr. 36). Pro srovnání lze správný výstupní obraz nalézt na obrázku (Obr. 37).



*Obr. 33. Testovací obraz ze soumrakové kamery*



*Obr. 34. Testovací obraz z IR kamery*



*Obr. 35. Uživatelem chybně zvolené body u algoritmu "Dva body"*



*Obr. 36. Chybně zvolené vstupní parametry u algoritmu "Transformace"*



*Obr. 37. Korektní výsledek sesouhlasení obrazů*

## ZÁVĚR

Zadáním diplomové práce bylo nastudovat algoritmy, které řeší fúzi obrazových dat dvou kamer pracujících na různých vlnových délkách a jeden z nich zrealizovat ve formě programu, který se otestuje na obrazových předlohách.

Byly vytvořeny dva algoritmy, které jsou schopny slučovat obrazy a oba tyto algoritmy byly zrealizovány ve formě programu. Z výsledků testování algoritmů vychází, že spolehlivější je algoritmus “Dva body“, ale přesnější je algoritmus “Transformace“. Kombinací algoritmů lze spojit tyto vlastnosti a dosáhnout zkvalitnění výsledků. Jednalo by se o postup, kdy si uživatel zvolí dva body na obou vstupních obrazech a z těchto bodů se vypočítají parametry, které vymezují oblast hledání sesouhlasení obrazů. Poté se provede algoritmus “Transformace“, který pracuje s těmito parametry.

Dalšími zlepšením programu může být:

- Automatické ořezání černých okrajů vyskytujících se u obrazů získaných z infračervených kamer. Zde by bylo použito ořezání na předem daný rozměr.
- Možnost výběru uživatelem, jestli se má provést algoritmus “Transformace“ s použitím detekce hran nebo bez něj. Záleželo by to na požadavku na přesnost sesouhlasení obrazů.
- Urychlení algoritmu “Transformace“ použitím ořezání obrazu při rotaci na původní velikost. Ovšem s větším úhlem rotace narůstá i možnost chybného sesouhlasení.
- Ošetření programu proti chybám ze strany uživatele, jako je například předčasné ukončení programu během výpočtu algoritmu.
- Přidání nápovědy do programu a možnosti ukládat výsledné hodnoty (změna měřítka, rotace a posuv) do textového souboru na disk.
- Propojením programu se dvěma kamerami a sesouhlasení obrazů získaných z těchto kamer v reálném čase.

To byl jen nástin možností zlepšení programu s tím, že hlavní část a to práce programu s kamerami v reálném čase by bylo vhodné řešit v návaznosti na tuto diplomovou práci.

## CONCLUSION

The main target of Master thesis was to elaborate an algorithm, which solve fusion of image data from two video-cameras working with different wavelength. This algorithm should be written as a program, which is tested on source images.

The two algorithms were created, so that they are able to combine images and both algorithms were made as program in MATLAB environment. An advantage of the first algorithm (Two points) is a speed and simplicity. The advantage of the second algorithm (Transformation) is accuracy. When both algorithms will be working together it is going to improve results.

Other improvement of developed program can be:

- Automatic cropping of black borders of images.
- The user can choose whether algorithm "Transformation" will be realized with edge detector or without edge detector.
- The image can be cropped when algorithm "Transformation" rotates the image. It accelerates the algorithm.
- Overcome bad inputs come from the user.
- Program will be include help
- Working with video-cameras in the real time.

It was summary of possibilities what can improve the program.

**SEZNAM POUŽITÉ LITERATURY**

- [1] *VOP-026-Šternberk* [online]. [2008] [cit. 2008-04-11]. Dostupný z WWW: <[http://www.vop.cz/vtuvm\\_slavicin\\_vyrobky\\_pos.htm](http://www.vop.cz/vtuvm_slavicin_vyrobky_pos.htm)>.
- [2] *Wikipedie: Elektromagnetické spektrum* [online]. 2008 , 29.3.2008 [cit. 2008-05-24]. Dostupný z WWW: <[http://cs.wikipedia.org/wiki/Elektromagnetick%C3%A9\\_spektrum](http://cs.wikipedia.org/wiki/Elektromagnetick%C3%A9_spektrum)>.
- [3] DOSTÁL, Jiří, JANÁČEK, Zdeněk. *FYZIKA*. 2. vyd. Zlín : UTB ve Zlíně, 2003. 265 s. ISBN 80-7318-115-0.
- [4] VOJÁČEK, Antonín. Co jsou BOLOMETRY a MIKROBOLOMETRY? *Automatizace.hw.cz : Měření a regulace* [online]. 2005 [cit. 2008-05-24]. Dostupný z WWW: <<http://automatizace.hw.cz/mereni-a-regulace/ART196-co-jsou-bolometry-a-mikrobolometry-.html>>.
- [5] *HUMUSOFT: MATLAB (popis produktu)* [online]. 2008 [cit. 2008-05-24]. Dostupný z WWW: <<http://www.humusoft.cz/matlab/matlab.htm>>.
- [6] *Image Processing Toolbox* [online]. 2008 [cit. 2008-05-24]. Dostupný z WWW: <<http://www.humusoft.cz/matlab/moduly/image.htm>>.
- [7] ZAPLATÍLEK, Karel, DOŇAR, Bohuslav. *MATLAB pro začátečníky*. 1. vyd. Praha : BEN – technická literatura, 2003. 114 s. ISBN 80-7300-095-4.
- [8] ZAPLATÍLEK, Karel, DOŇAR, Bohuslav. *MATLAB – tvorba uživatelských aplikací*. 1. vyd. Praha : BEN – technická literatura, 2004. 216 s. ISBN 80-7300-133-0.
- [9] *MATLAB* [online]. The MathWorks, Inc., c1984-2008 [cit. 2008-05-24]. Dostupný z WWW: <[http://www.mathworks.com/access/helpdesk/help/techdoc/creating\\_plots/f2-10709.html](http://www.mathworks.com/access/helpdesk/help/techdoc/creating_plots/f2-10709.html)>.
- [10] ČANDÍK, Marek. Metody estimace velikosti plochy snímaných objektů. In 29. *mezinárodní konference TD 2006 - DIAGON 2006*. Zlín : Academia centrum Univerzity Tomáše Bati ve Zlíně, 2006. s. 89-95. ISBN 80-7318-410-9.

- 
- [11] GONZALES, Rafael C., WOODS, Richard E., EDDINS, Steven L. *Digital image processing using MATLAB*. New Jersey : Pearson Education, Inc., c2004. 2 sv. (XIV, 609 s.). ISBN 0-13-008519-7.
- [12] ŽÁRA, Jiří, et al. *Moderní počítačová grafika*. 1. vyd. Brno : Computer Press, 2004. 609 s. ISBN 80-251-0454-0.
- [13] JIŘÍ JAN 2002. *Číslicová filtrace, analýza a restaurace signálů*. Brno: VUTIUM, 2002, 427 s. ISBN 80-214-1558-4.



**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

$\alpha$	Úhel rotace [°]
$\lambda$	Vlnová délka ve vakuu [m]
$f$	Frekvence kmitavého pohybu [Hz]
$c$	Rychlost šíření elektromagnetických vln (ve vakuu rovno 299 792 458 m/s)
$C_{a,b}$	Výstupní obraz diskrétní konvoluce
$f_{a-x,b-y}$	Vstupní statický obraz
$h_{a,b}$	Vstupní dynamický obraz
$a, b, x, y$	Souřadnice matice obrazů
$\rho_{a,b}$	Korelační koeficient v hodnotě $\langle -1,1 \rangle$
$\sigma_f^2$	Rozptyl statického obrazu
$\sigma_h^2$	Rozptyl dynamického obrazu
$R(\alpha)$	Transformační matice rotace
$S(s_x, s_y)$	Transformační matice změny měřítka
$s_x, s_y$	Koeficienty změny měřítka na ose x a ose y
CCD	Nábojově vázané fotocitlivé elementy ( <b>C</b> harge <b>C</b> oupled <b>D</b> evice)
IR	Infračervené záření ( <b>I</b> nfrared)
LoG	Hranový detektor <i>Laplacian of Gaussian</i>
POS	Lehký průzkumný a pozorovací systém
RGB	Barevný model RGB ( <b>R</b> ed <b>B</b> lue <b>G</b> reen)
2-D	Dvojdímenzionální prostor

## SEZNAM OBRÁZKŮ

Obr. 1. Lehký průzkumný a pozorovací systém POS [1].....	8
Obr. 2. Elektromagnetické spektrum [2].....	10
Obr. 3. Detail jedné plošky mikrobolometru [4].....	11
Obr. 4. Prostředí MATLAB .....	12
Obr. 5. Obraz a matice RGB [9] .....	13
Obr. 6. Obraz a matice stupnice šedi [9].....	13
Obr. 7. a) originální obraz; b) Sobel; c) Prewitt; d) Roberts; e) LoG; f) Canny .....	15
Obr. 8. Vlevo-původní velikost obrazu; vpravo-výsledný obraz po ořezání .....	17
Obr. 9. Sesouhlasení obrazů .....	20
Obr. 10. Načtení obrazů .....	21
Obr. 11. Varovné okno pro načítání.....	21
Obr. 12. Označení vybraných bodů hvězdou .....	22
Obr. 13. Nesprávně zadané body .....	22
Obr. 14. Obraz z kamery ve viditelné oblasti spektra.....	23
Obr. 15. Obraz z kamery v infračervené oblasti spektra.....	23
Obr. 16. Výsledný sesouhlasený obraz algoritmu Dva body .....	24
Obr. 17. Opravte vstupní parametry.....	25
Obr. 18. Zobrazení průběhu výpočtu .....	25
Obr. 19. Výsledný sesouhlasený obraz algoritmu Transformace.....	25
Obr. 20. Obraz z kamery ve viditelné oblasti spektra.....	26
Obr. 21. Obraz z kamery v infračervené oblasti spektra.....	26
Obr. 22. Uložení obrazu.....	27
Obr. 23. Varovné okno pro ukládání .....	27
Obr. 24. Ukončení programu .....	27
Obr. 25. Schématický náčrt vstupních obrazů.....	28
Obr. 26. Obraz z kamery ve viditelné oblasti spektra.....	29
Obr. 27. Obraz z kamery v infračervené oblasti spektra.....	29
Obr. 28. Obraz z kamery ve viditelné oblasti spektra.....	30
Obr. 29. Obraz z kamery v infračervené oblasti spektra.....	30
Obr. 30. Výsledek funkce <b>normxcrr2</b> .....	31
Obr. 31. Obraz z kamery v infračervené oblasti spektra.....	32

---

Obr. 32. Obraz z kamery v infračervené oblasti spektra.....	32
Obr. 33. Testovací obraz ze soumrakové kamery.....	34
Obr. 34. Testovací obraz z IR kamery .....	35
Obr. 35. Uživatelem chybně zvolené body u algoritmu “Dva body“.....	35
Obr. 36. Chybně zvolené vstupní parametry u algoritmu “Transformace“ .....	36
Obr. 37. Korektní výsledek sesouhlasení obrazů.....	36

## SEZNAM PŘÍLOH

P I Zdrojový kód programu **Sesouhlasení obrazů**

# PŘÍLOHA P I: ZDROJOVÝ KÓD PROGRAMU SESOUHLASENÍ OBRAZŮ

```
1 % - - - - - Jana Návrátová - - - - -
2 % Program pro sesouhlasení obrazů
3 % o různých vlnových délkách
4 %-----
5
6 function sesouhlaseni
7
8 clear all
9 close all
10
11 global hlavni informace1 informace2 informace3 informace4 monitor
    tl_nacti tl_uloz tl_body tl_transformace zoom_min zoom_max zoom_krok
    rotace_min rotace_max rotace_krok
12
13 monitor=get(0,'ScreenSize');
14 hlavni=figure('position',[(monitor(3)-205),...
15             (monitor(4)-530), 200, 500], 'resize', 'off',...
16             'numbertitle','off','name','Sesouhlasení','menubar',...
17             'none','color',[0.4 0.5 0.6],'CloseRequestFcn','konec', 'Bu-
    syAction', 'cancel');
18
19 tl_nacti=icontrol('Callback','nacti',...
20 'Position',[5 450 90 30],...
21 'Style','PushButton',...
22 'String','Načti obrázky',...
23 'style','push','FontSize',10);
24
25 tl_uloz=icontrol('Callback','uloz',...
26 'Position',[105 450 90 30],...
27 'Style','PushButton',...
28 'String','Ulož obraz',...
29 'style','push','Enable','off','FontSize',10);
30
31 tl_body=icontrol('Callback','body',...
32 'Position',[5 400 90 30],...
33 'Style','PushButton',...
34 'String','Označ body',...
35 'style','push','Enable','off','FontSize',10);
```

```
36 tl_transformace=uicontrol('Callback','transformace',...
37 'Position',[105 400 90 30],...
38 'Style','PushButton',...
39 'String','Transformuj',...
40 'style','push','Enable','off','FontSize',10);
41
42 informace1=uicontrol('Position',[5 125 190 20],...
43 'Style','Text','FontSize',8,'String','Výsledná změna měřítka:');
44
45 informace2=uicontrol('Position',[5 90 190 20],...
46 'Style','Text','FontSize',8,'String','Výsledná rotace:');
47
48 informace3=uicontrol('Position',[5 55 190 20],...
49 'Style','Text','FontSize',8,'String','Výsledný posuv v ose x:');
50
51 informace4=uicontrol('Position',[5 20 190 20],...
52 'Style','Text','FontSize',8,'String','Výsledný posuv v ose y:');
53
54 uicontrol('Position',[5 350 130 15],...
55 'Style','Text','FontSize',10,...
56 'String','Min. změna měřítka');
57
58 uicontrol('Position',[5 320 130 15],...
59 'Style','Text','FontSize',10,...
60 'String','Max. změna měřítka');
61
62 uicontrol('Position',[5 290 130 15],...
63 'Style','Text','FontSize',10,...
64 'String','Krok změny');
65
66 zoom_min=uicontrol('Position',[145 350 50 20],'String','0.9',...
67 'Style','Edit','BackgroundColor','white');
68
69 zoom_max=uicontrol('Position',[145 320 50 20],'String','1.1',...
70 'Style','Edit','BackgroundColor','white');
71
72 zoom_krok=uicontrol('Position',[145 290 50 20],'String','0.1',...
73 'Style','Edit','BackgroundColor','white');
74
75 uicontrol('Position',[5 250 130 15],...
76 'Style','Text','FontSize',10,...
```

```

77 'String','Minimální rotace');
78
79 uicontrol('Position',[5 220 130 15],...
80 'Style','Text','FontSize',10,...
81 'String','Maximální rotace');
82
83 uicontrol('Position',[5 190 130 15],...
84 'Style','Text','FontSize',10,...
85 'String','Krok rotace');
86
87 rotace_min=uicontrol('Position',[145 250 50 20],'String','-1',...
88     'Style','Edit','BackgroundColor','white');
89
90 rotace_max=uicontrol('Position',[145 220 50 20],'String','1',...
91     'Style','Edit','BackgroundColor','white');
92
93 rotace_krok=uicontrol('Position',[145 190 50 20],'String','1',...
94     'Style','Edit','BackgroundColor','white');
95
96 end

97 %--načtení obrazů--
98
99 function nacti
100 global monitor FileName1 FileName2 obr1 obr2 zobraz1 zobraz2 tl_body
    tl_transformace
101
102 [FileName1,PathName1] = uigetfile({'*.bmp;*.jpg','Obrazy
    (*.bmp,*.jpg)'},'Načtení obrazů');
103
104 if FileName1~=0
105     [FileName2,PathName2] = uigetfile({'*.bmp;*.jpg','Obrazy
    (*.bmp,*.jpg)'},'Načtení obrazů');
106     if FileName2~=0
107         iptsetpref('ImshowBorder','tight')
108         p_obr1=imread([PathName1,FileName1]);
109         p_obr2=imread([PathName2,FileName2]);
110         p_O = size(p_obr1,1);
111         p_P = size(p_obr1,2);
112         p_W = size(p_obr2,1);
113         p_V = size(p_obr2,2);

```

```

114     p_Q = p_O*p_P;
115     p_U = p_W*p_V;
116
117     if p_Q < p_U
118         obr2 = p_obr1;
119         obr1 = p_obr2;
120     else
121         obr1 = p_obr1;
122         obr2 = p_obr2;
123     end
124
125     zobraz1=figure('position',[monitor(3)-size(obr1,2))/2-
50,(monitor(4)-size(obr1,2))/2+50, size(obr1,2), size(obr1,1)], ...
126         'resize', 'of-
f','numbertitle','off','name',FileName1,'menubar','none');
127     imshow(obr1)
128     zobraz2=figure('position',[monitor(3)-
size(obr2,2))/2+50,(monitor(4)-size(obr2,2))/2-50, size(obr2,2), si-
ze(obr2,1)],...
129         'resize', 'of-
f','numbertitle','off','name',FileName2,'menubar','none');
130     imshow(obr2)
131
132     set(tl_body,'Enable','on');
133     set(tl_transformace,'Enable','on');
134
135     else
136         errordlg('Chyba názvu souboru','Soubor nenalezen')
137     end
138 else
139     errordlg('Chyba názvu souboru','Soubor nenalezen')
140 end
141
142 end

143 % uložení výsledného obrazu na disk
144
145 function uloz
146 global obr_soucet
147
148 [FileName3,PathName3] = uiputfile({'*.bmp;*.jpg','Obrázky
(*.bmp,*.jpg)'),'Uložení obrazu','vysledek.bmp');

```



```

149
150 if FileName3~=0
151     imwrite(obr_soucet,[PathName3,FileName3]);
152 else
153     errordlg('Chyba názvu souboru','Soubor nelze uložit')
154 end
155
156 end

157 %--algoritmus Dva body--
158
159 function body
160 global monitor informace1 informace2 informace3 informace4 zobraz1
    zobraz2 obr1 obr2 obr_soucet tl_nacti tl_uloz tl_body
    tl_transformace
161
162 try
163
164 set(tl_nacti,'Enable','off')
165 set(tl_uloz,'Enable','off')
166 set(tl_body,'Enable','off')
167 set(tl_transformace,'Enable','off')
168
169 % --určení souřadnic bodů--
170
171 figure (zobraz1)
172 [sloup1,rad1,RGB1] = impixel(obr1);
173 figure (zobraz2)
174 [sloup2,rad2,RGB2] = impixel(obr2);
175
176 % --výpočet úseček--
177
178 sloupec1=sloup1(2)-sloup1(1);
179 radek1=rad1(1)-rad1(2);
180 sloupec2=sloup2(2)-sloup2(1);
181 radek2=rad2(1)-rad2(2);
182 vzdal1=sqrt(sloupec1*sloupec1+radek1*radek1);
183 vzdal2=sqrt(sloupec2*sloupec2+radek2*radek2);
184
185 % --výpočet změny měřítka--
186

```

```

187 xxx=vzdal1/vzdal2;
188
189 % --výpočet rotace--
190
191 uhel1=atan(radek1/sloupec1);
192
193 if sloup1(2)<sloup1(1)
194     uhel1=uhel1+pi;
195 end
196
197 uhel2=atan(radek2/sloupec2);
198
199 if sloup2(2)<sloup2(1)
200     uhel2=uhel2+pi;
201 end
202
203 yyy=(uhel1-uhel2)*180/pi;
204
205 % --výpočet posunutí bodů--
206
207 s_s=size(obr2,2);
208 s_r=size(obr2,1);
209
210 matice=zeros(s_r,s_s);
211 matice(rad2(1),sloup2(1))=1;
212
213 zmena = imresize(matice,xxx,'bicubic');
214 pomoc = imrotate(zmena,yyy,'bicubic');
215
216 abc = max(pomoc);
217 maximum = max(abc);
218 [rd,sl] = find(pomoc==maximum);
219
220 sour_s=sloup1(1)-sl;
221 sour_r=rad1(1)-rd;
222
223 %--otočení a změna obrazu--
224
225 zmena2 = imresize(obr2,xxx,'bicubic');
226 obr3 = imrotate(zmena2,yyy,'bicubic');
227

```

```
228 %--překreslení výsledného obrazu--
229
230 sss_s=round(sour_s)+1;
231 rrr_r=round(sour_r)+1;
232
233 if sss_s>0
234     if size(obr1,2)>(size(obr3,2)+sss_s)
235         Zx=size(obr1,2);
236     else
237         Zx=size(obr3,2)+sss_s-1;
238     end
239 else
240     if size(obr3,2)>(size(obr1,2)-sss_s)
241         Zx=size(obr3,2);
242     else
243         Zx=size(obr1,2)-sss_s;
244     end
245 end
246
247 if rrr_r>0
248     if size(obr1,1)>(size(obr3,1)+rrr_r)
249         Zy=size(obr1,1);
250     else
251         Zy=size(obr3,1)+rrr_r-1;
252     end
253 else
254     if size(obr3,1)>(size(obr1,1)-rrr_r)
255         Zy=size(obr3,1);
256     else
257         Zy=size(obr1,1)-rrr_r;
258     end
259 end
260
261 matice=zeros(Zy,Zx,3);
262
263 if sss_s==0
264     sss_s=1;
265 end
266
267 if rrr_r==0
268     rrr_r=1;
```

```

269 end
270
271
272 if sss_s>=0
273     prlx=1;
274     pr2x=sss_s;
275 else
276     prlx=-sss_s;
277     pr2x=1;
278 end
279
280 if rrr_r>=0
281     prly=1;
282     pr2y=rrr_r;
283 else
284     prly=-rrr_r;
285     pr2y=1;
286 end
287
288 obr1d=double(obr1);
289 obr3d=double(obr3);
290 matice(prly:(prly+size(obr1d,1)-1),prlx:(prlx+size(obr1d,2)-
    1),1:3)=0.5*obr1d(1:size(obr1d,1),1:size(obr1d,2),1:3);
291 matice(pr2y:(pr2y+size(obr3d,1)-1),pr2x:(pr2x+size(obr3d,2)-
    1),1:3)=0.5*obr3d(1:size(obr3d,1),1:size(obr3d,2),1:3)+matice(pr2y:(
    pr2y+size(obr3d,1)-1),pr2x:(pr2x+size(obr3d,2)-1),1:3);
292 obr_soucet=uint8(matice);
293
294 figure('position',[ (monitor(3)-size(matice,2))/2, (monitor(4)-
    size(matice,2))/2, size(matice,2), size(matice,1)],...
295     'numbertitle','off','name','Výsledný obraz','menubar',...
296     'none','resize','off');
297 imshow(obr_soucet)
298
299 set(tl_nacti,'Enable','on')
300 set(tl_uloz,'Enable','on')
301 set(tl_body,'Enable','on')
302 set(tl_transformace,'Enable','on')
303
304 set(informace1,'String',['Výsledná změna měřítka: ',num2str(xxx)]);
305 set(informace2,'String',['Výsledná rotace: ',num2str(yyy)]);
306

```

```

307 if prlx>pr2x
308 set(informace3,'String',['Výsledný posuv v ose x: ',num2str(prlx)]);
309 else
310 set(informace3,'String',['Výsledný posuv v ose x: ',num2str(pr2x)]);
311 end
312
313 if prlx>pr2x
314 set(informace4,'String',['Výsledný posuv v ose y: ',num2str(prly)]);
315 else
316 set(informace4,'String',['Výsledný posuv v ose y: ',num2str(pr2y)]);
317 end
318
319 catch
320   errordlg('Nesprávně zadané body','Pozor chyba')
321   set(tl_nacti,'Enable','on')
322   set(tl_body,'Enable','on')
323   set(tl_transformace,'Enable','on')
324 end
325
326 end
327
328

329 %--algoritmus Transformace--
330
331 function transformace
332   global monitor ukazatel h informace1 informace2 informace3 informa-
   ce4 zoom_min zoom_max zoom_krok rotace_min rotace_max rotace_krok
   obr1 obr2 obr_soucet tl_nacti tl_uloz tl_body tl_transformace
333
334 try
335
336 %--načtení parametrů--
337
338 min_z=str2double(get(findobj(zoom_min),'String'));
339 max_z=str2double(get(findobj(zoom_max),'String'));
340 krok_z=str2double(get(findobj(zoom_krok),'String'));
341 min_r=str2double(get(findobj(rotace_min),'String'));
342 max_r=str2double(get(findobj(rotace_max),'String'));
343 krok_r=str2double(get(findobj(rotace_krok),'String'));
344

```

```

345 set(tl_nacti, 'Enable', 'off')
346 set(tl_uloz, 'Enable', 'off')
347 set(tl_body, 'Enable', 'off')
348 set(tl_transformace, 'Enable', 'off')
349
350 %--převedení obrazů do stupnice šedi a vyhledání hran--
351
352 sed1 = rgb2gray (obr1);
353 sed2 = rgb2gray (obr2);
354
355 sedy1 = edge(sed1, 'canny');
356 sedy2 = edge(sed2, 'canny');
357
358 k = 1;
359 prom_max = -1;
360 zaloha=sedy1;
361 pocet_cyklu=0;
362 ukazatel=0;
363
364 for zoom_konst = min_z: krok_z: max_z
365     for rotac_konst = min_r: krok_r: max_r
366         pocet_cyklu=1+pocet_cyklu;
367     end
368 end
369
370 h = waitbar(0, 'Průběh transformace');
371
372 %--cyklus otáčení a změny měřítka obrazu--
373
374 for zoom_konst = min_z: krok_z: max_z
375     zoom_obr = imresize(sedy2, zoom_konst, 'bicubic');
376
377     for rotac_konst = min_r: krok_r: max_r
378         rotac_obr = imrotate(zoom_obr, rotac_konst, 'bicubic');
379
380         O=size(rotac_obr,1);
381         P=size(rotac_obr,2);
382
383         sedy1=zaloha;
384         W = size(sedy1,1);
385         V = size(sedy1,2);

```

```

386
387     if V<P
388         sedyl(1:W,V+1:P)=zeros;
389     end
390
391     V = size(sedyl,2);
392
393     if W<0
394         sedyl(W+1:0,1:V)=zeros;
395     end
396
397 %--výpočet diskrétní konvoluce a korelačního koeficientu--
398
399     C = normxcorr2(double(rotac_obr),double(sedyl));
400
401     D = abs(C);
402     abc = max(D);
403     maximum(k) = max(abc);
404
405     if maximum(k)>prom_max
406         prom_max=maximum(k);
407         prom_zoom=zoom_konst;
408         prom_rotac=rotac_konst;
409         [rd,sl] = find(C==maximum(k));
410     end
411
412     waitbar(k/pocet_cyklu,h)
413
414     k=k+1;
415
416     end
417 end
418
419 close(h)
420
421 %--otočení a změna obrazu--
422
423 zmena = imresize(obr2,prom_zoom,'bicubic');
424 obr3 = imrotate(zmena,prom_rotac,'bicubic');
425
426 %--překreslení obrazu--

```

```

427 Cx=s1;
428 Cy=rd;
429 Dx=size(obr3,2);
430 Dy=size(obr3,1);
431 Sx=size(obr1,2);
432 Sy=size(obr1,1);
433
434 if Cx < Dx
435     pr2x=1;
436     pr1x=Dx-Cx+1;
437     Zx=max(pr1x+Sx-1,Dx);
438 else
439     pr1x=1;
440     pr2x=Cx-Dx+1;
441     Zx=max(pr2x+Dx-1,Sx);
442 end
443
444 if Cy < Dy
445     pr2y=1;
446     pr1y=Dy-Cy+1;
447     Zy=max(pr1y+Sy-1,Dy);
448 else
449     pr1y=1;
450     pr2y=Cy-Dy+1;
451     Zy=max(pr2y+Dy-1,Sy);
452 end
453
454
455 matice=zeros(Zy,Zx,3);
456 obr1d=double(obr1);
457 obr3d=double(obr3);
458 matice(pr1y:(pr1y+Sy-1),pr1x:(pr1x+Sx-1),1:3)=0.5*obr1d(1:Sy,1:Sx,1:3);
459 matice(pr2y:(pr2y+Dy-1),pr2x:(pr2x+Dx-1),1:3)=0.5*obr3d(1:Dy,1:Dx,1:3)+matice(pr2y:(pr2y+Dy-1),pr2x:(pr2x+Dx-1),1:3);
460
461 obr_soucet=uint8(matice);
462 figure('position',[monitor(3)-size(matice,2))/2,(monitor(4)-size(matice,2))/2, size(matice,2), size(matice,1)],...
463     'numbertitle','off','name','Výsledný obraz','menubar',...
464     'none','resize','off');

```



```

465 imshow(obr_soucet)
466
467 set(informace1,'String',['Výsledná změna měřítka:
    ',num2str(prom_zoom)]);
468 set(informace2,'String',['Výsledná rotace: ',num2str(prom_rotac)]);
469
470 if prlx>pr2x
471 set(informace3,'String',['Výsledný posuv v ose x: ',num2str(prlx)]);
472 else
473 set(informace3,'String',['Výsledný posuv v ose x: ',num2str(pr2x)]);
474 end
475 if prlx>pr2x
476 set(informace4,'String',['Výsledný posuv v ose y: ',num2str(pr1y)]);
477 else
478 set(informace4,'String',['Výsledný posuv v ose y: ',num2str(pr2y)]);
479 end
480
481 set(tl_uloz,'Enable','on')
482
483 catch
484     errordlg('Opravte vstupní parametry','Pozor chyba')
485 end
486
487 set(tl_nacti,'Enable','on')
488 set(tl_body,'Enable','on')
489 set(tl_transformace,'Enable','on')
490
491 end

492 %--ukončení programu--
493
494 function konec
495     global hlavni
496
497         delete(hlavni)
498         close all
499         clear all
500         clc
501
502 end

```