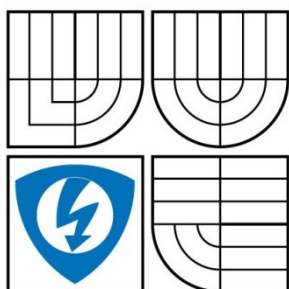


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKACNÍCH
TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

ÚPRAVA OBECNÉHO STEREO PÁRU OBRAZU DO JEDNODUCHÉ STEREO GEOMETRIE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

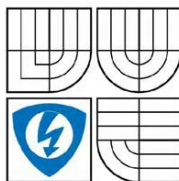
AUTOR PRÁCE
AUTHOR

JAN PELOUŠEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. KAMIL ŘÍHA, Ph.D.

BRNO 2009



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Jan Peloušek
Ročník: 3

ID: 72753
Akademický rok: 2008/2009

NÁZEV TÉMATU:

Úprava obecného stereo páru obrazů do jednoduché stereo geometrie

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte teoretický postup při úpravě obrazů sejmutých dvojicí totožných kamer v obecné pozici na snímky spojené jednoduchou stereo geometrií (model dvojice paralelních kamer) na základě ručně zadaných korespondencí. Tento postup realizujte formou samostatné aplikace.

DOPORUČENÁ LITERATURA:

- [1] GONZALEZ R. C., WOODS R. E.: Digital Image Processing, Prentice Hall, New Jersey, 2002
- [2] FISHER R. B.: CVonline: The Evolving, Distributed, Non-Proprietary, On-Line Compendium of Computer Vision, <http://homepages.inf.ed.ac.uk/rbf/CVonline/>
- [3] HARTLEY R., ZISSERMANN A.: Multiple View Geometry in Computer Vision. Cambridge: Cambridge University Press, 2003.

Termín zadání: 9.2.2009

Termín odevzdání: 2.6.2009

Vedoucí práce: Ing. Kamil Říha, Ph.D.

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

PODĚKOVÁNÍ

Děkuji vedoucímu semestrální práce Ing. Kamilu Říhovi, PhD., za velmi užitečnou metodickou pomoc a cenné rady při zpracování mé bakalářské práce.

V Brně dne

Podpis

ANOTACE:

Tato práce pojednává o základech epipolární geometrie, vysvětluje způsob automatické detekce korespondujících bodů, popisuje principy tvorby fundamentální matice a konečně tvorby mezilehlého pohledu. Dále jsou zde popsány některé základní funkce knihovny OpenCV od firmy Intel, které s danou problematikou souvisí, potažmo popisuje způsob počítačového zpracování stereo-páru. Součástí práce je také samostatná aplikace, vytvořená v jazyce C++, konkrétně ve vývojovém prostředí Borland C++ Builder 6.0 za pomoci knihovny OpenCV. Tato aplikace umožňuje demonstrovat některé zde uvedené pojmy. Zejména potom výpočet fundamentální matice a tvorbu mezilehlého pohledu. V souvislosti s použitím prostředí Borland C++ Builder 6.0 je zde také popsán způsob integrace OpenCV do tohoto prostředí.

KLÍČOVÁ SLOVA:

stereogeometrie, fundamentální matice, epipolární geometrie, epipolární přímka, epipolární rovina, epipól, korespondence, rektifikace, OpenCV, disparita, morphing mezilehlého pohledu, SIFT

ABSTRACT:

This Bachelor's thesis considers the basics of epipolar geometry, explains the process of the automatic correspondence detection, describes principals of the fundamental matrix computation and finally the virtual camera view morphing between two images. There were described some of the basic Intel's OpenCV library functions joined with this issue and describes de facto the computer stereo-pair processing. As a part of this thesis, there is the application, programmed in the C++ language in the Borland C++ Builder 6.0 developer's environment by using the OpenCV library. This application demonstrates some of here-mentioned terms such as the fundamental matrix computation and the virtual camera view morphing. In relation to the Borland C++ Builder 6.0 using there was described the integration method of OpenCV to this environment.

KEYWORDS:

stereogeometry, fundamental matrix, epipolar geometry, epiline, epipolar line, epipolar plane, epipole, correspondence, rectification, OpenCV, disparity, internal view morphing, SIFT

CITACE:

PELOUŠEK, J. *Úprava obecného stereo páru obrazů do jednoduché stereo geometrie*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 31 s. Vedoucí bakalářské práce Ing. Kamil Říha, Ph.D.

OBSAH:

ÚVOD:	1
1. EPIPOLÁRNÍ GEOMETRIE	1
1.1 DISPARITA	3
1.2 FUNDAMENTÁLNÍ MATICE A JEJÍ VLASTNOSTI	3
1.2.1 Základní způsob určení fundamentální matice	4
1.2.2 Přesun bodu pomocí roviny Π	4
1.2.3 Konstrukce epipolární přímky	5
1.2.4 Algoritmy pro výpočet fundamentální matice	6
2. JEDNODUCHÁ STEREOGEOMETRIE	8
2.1 REKTIFIKACE OBRÁZKŮ	8
3. MORPHING MEZILEHLÉHO POHLEDU	9
4. POČÍTAČOVÉ ZPRACOVÁNÍ STEREO-PÁRU	11
4.1 NAČTENÍ STEREO-PÁRU DO PAMĚTI A PŘÍPRAVA PRO PRÁCI	12
4.1.1 Načtení do paměti	12
4.1.2 Vytvoření okna	12
4.1.3 Zobrazení obrázků	12
4.2 ZADÁVÁNÍ KORESPONDENCÍ	13
4.2.1 Ruční zadávání	12
4.2.2 Automatické hledání korespondencí pomocí algoritmu SIFT	14
4.3 VÝPOČET FUNDAMENTÁLNÍ MATICE	16
4.4 ROZDĚLENÍ OBRÁZKŮ NA ŘÁDKY	17
4.5 REKTIFIKACE OBRÁZKŮ	17

4.6 ROZDĚLENÍ ŘÁDKŮ NA BĚHY	17
4.7 HLEDÁNÍ KORESPONDENCÍ MEZI BĚHY	18
4.8 VYTVOŘENÍ ŘÁDKOVÁNÍ VIRTUÁLNÍ KAMERY	18
4.9 MORPHING OBRÁZKŮ	18
4.10 VYTVOŘENÍ VÝSLEDNÉHO POHLEDU VIRTUÁLNÍ KAMERY	18
4.11 VÝSLEDNÉ VYHLAZENÍ	19
 5. STRUČNÝ POPIS PROGRAMU StereoG	20
 6. VÝSLEDKY TVORBY MEZILEHLÉHO POHLEDU POMOCÍ PROGRAMU StereoG A NEDOSTATKY	23
 7. IMPLEMENTACE KNIHOVEN OpenCV DO BORLAND C++ BUILDERU ..	25
 ZÁVĚR	29
 POUŽITÁ LITERATURA	30
 POPIS PŘÍLOHY	31

Seznam obrázků:

Obr.1.: Binokulární vidění	1
Obr.2.: Epipolární geometrie	2
Obr.3.: Příklad disparitní mapy, vypočítané aplikací StereoG	3
Obr.4.: Přesun bodu pomocí roviny Π	5
Obr.5.: Jednoduchá stereogeometrie	8
Obr.6.: Hledání lokálních extrémů	15
Obr.7.: Vzhled programu StereoG	15
Obr.8.: Výsledky tvorby mezilehlého pohledu pomocí aplikace StereoG	23

Ostatní obrázky (kapitola 7) jsou pouze ilustrační

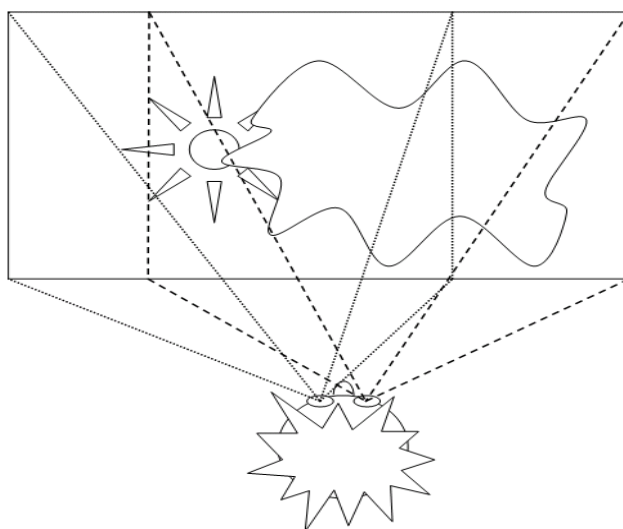
ÚVOD:

Tato práce se bude zabývat geometrií trojrozměrné scény, nasnímané dvěma identickými, umístěním nezávislými kamerami. Velká část práce se bude zabývat epipolární geometrií, která tuto problematiku popisuje.

Přílohou práce bude samostatná aplikace, která jednotlivé principy bude demonstrovat.

Důvodem, proč se zabývat stereogeometrií je matematické popsání lidského vidění. Každé z našich dvou očí je umístěno na odlišném místě hlavy. Jedno oko vidí obraz poněkud odlišný, o jistou vzdálenost posunutý oproti druhému. Tyto obrazy jsou dvourozměrné, ovšem člověk se zdravým binokulárním viděním (dvěma očima) vidí obraz jako jediný, prostorový. Právě ona odlišnost dvou obrazů vytváří vjem hloubky prostoru. Geometrické základy stereoskopického vidění popsal roku 1838 Sir Charles Wheatstone, který sestrojil první stereoskop a za svůj výzkum získal roku 1940 ocenění Královské vědecké společnosti.

Technicky můžeme lidské oko nahradit kamerou, popřípadě fotoaparátem pro získání dvourozměrných obrazů. Naším problémem tedy pro modelování počítačového vidění bude rekonstrukce trojrozměrného obrazu ze dvou obrazů dvourozměrných. Dvojici takovýchto obrazů budeme nazývat *stereo-pár*. Prostorovou informaci je možno na základě korespondujících si bodů v obou scénách možné popsat pomocí epipolární geometrie a na základě této prostorové informace je možné vytvářet i nové pohledy, nacházející se mezi pohledy reálnými.



Obr.1.: Binokulární vidění

1.EPIPOLÁRNÍ GEOMETRIE

Epipolární geometrie je základní geometrií virtuálního pohledu mezi dvěma pohledy reálnými. Není závislá na kompozici scény, pouze popisuje relativní polohu dvou kamer a jejich vnitřní parametry[2].

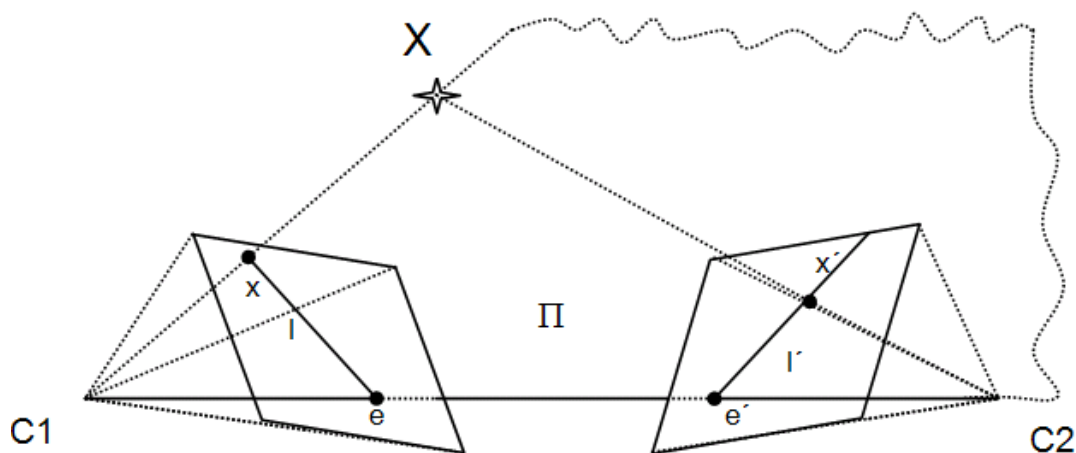
[2]: Epipolární geometrie mezi dvěma pohledy je v podstatě geometrií průniku roviny obrázku se svazkem rovin, majícími osu na spojnici středů obou kamer. Uvažujme bod X v trojrozměrném prostoru, který je v prvním obrázku zobrazen jako x a ve druhém jako x' . Jak je vidět na obr.1, tvoří body X , x , x' jednu rovinu a jsou koplanární (leží v jedné rovině) i s kamerovými středy. Označíme tuto rovinu, jako Π . Paprsky, promítnuté z x a x' protínající se v X také leží v Π . To je nejdůležitější vlastnost pro hledání korespondencí.

Zavedeme nové pojmy, definované ve[2]:

Epipól: Je bod průniku přímky spojující středy obou kamer s rovinou obrázku.

Epipolární rovina: Je rovina, obsahující spojnici kamerových středů. Je tedy takto definován jednoparametrový svazek epipolárních rovin.

Epipolární přímka: Je průnik epipolární roviny s rovinou obrazu. Všechny epipolární přímky se protínají v epipólu. Epipolární rovina protíná levý a pravý obrázek v epipolárních přímkách a definuje korespondenci mezi přímkami.



Obr.2: Epipolární geometrie

1.1 DISPARITA:

Máme-li stereo-pár, ve kterém jsou totožné reálné kamery od sebe nenulově posunuty, pak odpovídající si body v každém ze snímků jsou od sebe také určitým způsobem vzdáleny. Právě rozdílnost zobrazení korespondujících bodů do obrazové roviny se nazývá disparita. Je standardně definována trojrozměrným vektorem (posunutí v ose x , v ose y a velikost posunutí). V případě rektifikovaných obrázků (viz. kapitola 2.1.) se nám vektor redukuje o vertikální směr posunutí. Analýza disparit se provádí nejčastěji v úrovni jasových složek obrazu. Jestliže vyjádříme tyto rozdíly pro všechny obrazové body, je možné je zanást do tzv. disparitní mapy.



Obr.3.: Příklad disparitní mapy vypočítané aplikací StereoG

1.2 FUNDAMENTÁLNÍ MATICE A JEJÍ VLASTNOSTI:

Podle [2] je algebraickým vyjádřením epipolární geometrie, mapuje polohu bodu v jednom obraze na epipolární přímku ve druhém obraze. Jestliže máme bod v jednom obraze, pak existuje k němu korespondující epipolární přímka v obraze druhém. Každý bod $\mathbf{x}'$ druhého obrazu, odpovídající bodu $\mathbf{x}$ obrazu prvního musí ležet na epipolární přímce l' . Epipolární přímka je projekcí (ve druhém obraze) paprsku, procházejícího bodem $\mathbf{x}$ a středem kamery $\mathbf{C}$ první kamery. Potom lze mapovat $\mathbf{x} \rightarrow l'$

z jednoho bodu v obraze na jeho korespondující epipolární přímku v obraze druhém.

Vlastnosti fundamentální matice podle [1]:

1. Pro libovolný bod $\mathbf{x}$ v prvním obraze existuje odpovídající epipolární přímka $\mathbf{l} = \mathbf{F}\mathbf{x}$. Obdobně pro bod $\mathbf{x}'$ ve druhém obraze platí $\mathbf{l} = \mathbf{F}^T\mathbf{x}'$.
2. Pro libovolný bod $\mathbf{x}$ různý od $\mathbf{e}$ obsahuje epipolární přímka $\mathbf{l} = \mathbf{F}^T\mathbf{x}'$ epipól $\mathbf{e}'$. Platí tedy $\mathbf{e}'^T(\mathbf{F}\mathbf{x}) = (\mathbf{e}'^T\mathbf{F})\mathbf{x} = 0$ pro všechna $\mathbf{x}$. Musí tedy platit i $\mathbf{e}'^T = 0$ a $\mathbf{F}\mathbf{e} = 0$.
3. Fundamentální matice podle [1] má nulovou hodnotu determinantu, proto má pouze sedm stupňů volnosti oproti obecné homogenní matici s rozměry 3x3, která jich má osm.

1.2.1 Základní způsob určení fundamentální matice F

Předpokládáme, že již máme soubor korespondujících bodů. Jestliže máme souřadnice bodu $C[x_l, y_l]$ a bodu $C'[x_r, y_r]$. Za použití fundamentální matice F je možné podle [8] psát:

$$(x_l, y_l, 1) \cdot \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix} \cdot \begin{pmatrix} x_r \\ y_r \\ 1 \end{pmatrix} = 0$$

Tím získáme soustavu osmi lineárních rovnic:

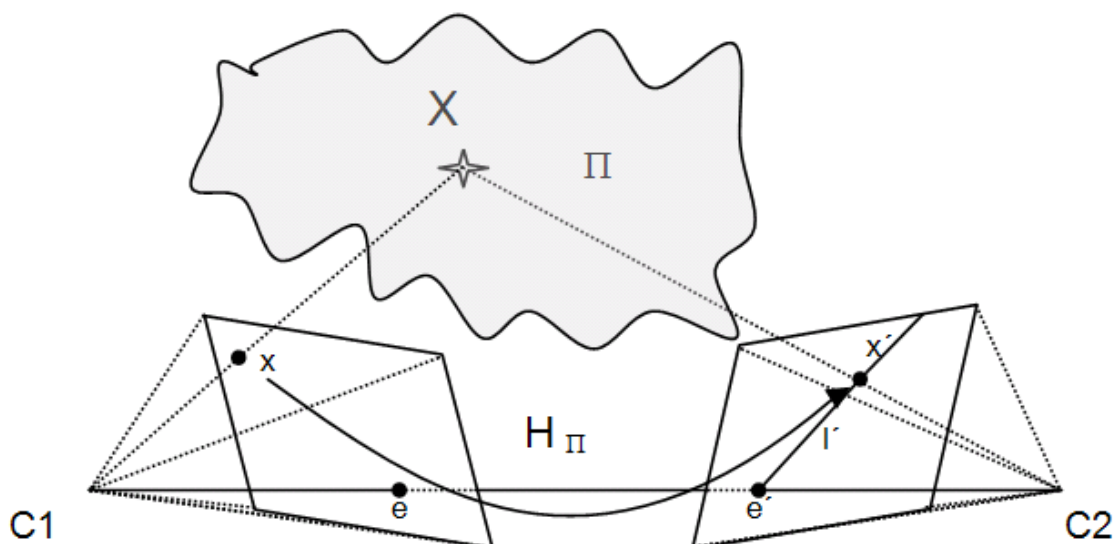
$$\begin{pmatrix} x_{l1} \cdot x_{r1} & x_{l1} \cdot y_{r1} & x_{l1} & y_{l1} \cdot x_{r1} & y_{l1} \cdot y_{r1} & y_{l1} & x_{r1} & y_{r1} \\ x_{l2} \cdot x_{r2} & x_{l2} \cdot y_{r2} & x_{l2} & y_{l2} \cdot x_{r2} & y_{l2} \cdot y_{r2} & y_{l2} & x_{r2} & y_{r2} \\ x_{l3} \cdot x_{r3} & x_{l3} \cdot y_{r3} & x_{l3} & y_{l3} \cdot x_{r3} & y_{l3} \cdot y_{r3} & y_{l3} & x_{r3} & y_{r3} \\ x_{l4} \cdot x_{r4} & x_{l4} \cdot y_{r4} & x_{l4} & y_{l4} \cdot x_{r4} & y_{l4} \cdot y_{r4} & y_{l4} & x_{r4} & y_{r4} \\ x_{l5} \cdot x_{r5} & x_{l5} \cdot y_{r5} & x_{l5} & y_{l5} \cdot x_{r5} & y_{l5} \cdot y_{r5} & y_{l5} & x_{r5} & y_{r5} \\ x_{l6} \cdot x_{r6} & x_{l6} \cdot y_{r6} & x_{l6} & y_{l6} \cdot x_{r6} & y_{l6} \cdot y_{r6} & y_{l6} & x_{r6} & y_{r6} \\ x_{l7} \cdot x_{r7} & x_{l7} \cdot y_{r7} & x_{l7} & y_{l7} \cdot x_{r7} & y_{l7} \cdot y_{r7} & y_{l7} & x_{r7} & y_{r7} \\ x_{l8} \cdot x_{r8} & x_{l8} \cdot y_{r8} & x_{l8} & y_{l8} \cdot x_{r8} & y_{l8} \cdot y_{r8} & y_{l8} & x_{r8} & y_{r7} \end{pmatrix} \cdot \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{pmatrix}$$

Řešením této soustavy lineárních rovnic dostaneme fundamentální matici F s prvky F_{11} až F_{33} , uspořádanými, jak je ukázáno výše.

1.2.2 Přesun bodu pomocí roviny Π podle[2]:

Jako na Obr.2 uvažujme rovinu Π , neprocházející žádným z kamerových středů. Přímka procházející prvním středem kamery a bodem $\mathbf{x}$ „narazí“ na

rovinu Π v bodě X . Tento bod je poté promítnut na bod x' ve druhém obrázku. Tento postup je známý jako „Přesun bodu pomocí roviny Π “. Jestliže bod X leží na přímce, odpovídající x , promítnutý bod x' musí ležet na epipolární přímce l' , odpovídající obrázku této přímky. Body x a x' jsou obrazy 3D bodu X ležící v rovině. Množina takových bodů x_i v prvním obrázku a korespondující bod x'_i ve druhém je ekvivalentní, co se týče projekce, rovinnému bodu X_i .



Obr.4: Přesun bodu pomocí roviny Π

Bod x v jednom obrázku je přesouván pomocí roviny Π na příslušný bod x' ve druhém obrázku. Epipolární přímku, procházející bodem x' získáme jeho spojením s epipólem e' . Symbolicky můžeme zapsat $x' = H_{\Pi}x$

1.2.3 Konstrukce epipolární přímky podle [2]:

Jestliže epipolární přímka l' prochází bodem x' a epipólem e' , můžeme napsat $l' = e' \times x' = [e']_{\times} x'$ a jestliže můžeme bod x' popsat, jako $x' = H_{\Pi}x$, dostaneme $l' = [e']_{\times} H_{\Pi}x = Fx$ a můžeme definovat fundamentální matici $F = [e']_{\times} H_{\Pi}$

pozn.: Jestliže je bod e' reprezentován trojrozměrným vektorem $e' = (x, y, z)^T$, pak

$$[e']_{\times} = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}$$

Rozměry fundamentální matice jsou podle[2] 3x3, (popřípadě 9x9, což jsou 3 matice o rozměrech 3x3 v případě sedmibodového algoritmu) s hodnotí 2.

1.2.4 Algoritmy pro výpočet fundamentální matice

Algoritmy pro stanovení fundamentální matice můžeme rozdělit do tří skupin:

- 1.lineární (pomocí řešení soustavy lineárních rovnicí)
- 2.iterační (pomocí opakování nějakého úkonu)
- 3.robustní (pro velké množství vstupních dat)

Její stanovení je možné pomocí přibližně 19 různých algoritmů, pro zajímavost uvádím seznam anglických, běžně užívaných, názvů algoritmů podle [5]:

(1) seven points; (2) least-squares (LS); (3) orthogonal LS; (4) rank-2 constraint; (5) iterative linear; (6) iterative Newton–Raphson; (7) minimization in parameter space; (8) gradient using LS; (9) gradient using eigen; (10) FNS; (11) CFNS; (12) M-Estimator using LS; (13) M-Estimator using eigen; (14) M-Estimator proposed by Torr; (15) LMedS using LS; (16) LMedS using eigen; (17) RANSAC; (18) MLESAC; (19) MAPSAC.

Nejběžněji používané jsou 4:

- 7 - bodový algoritmus: Petřička mezi lineární metody. Jejím řešením jsou až tři matice o rozměrech 3x3. Jak již název napovídá je k jeho výpočtu potřeba sedm dvojic korespondencí a jedná se o řešení soustavy lineárních rovnic.
- 8 - bodový algoritmus: Lineární algoritmus. Nejjednodušší metoda výpočtu fundamentální matice, skládající se z vytvoření a řešení soustavy lineárních rovnic. Pro úspěšné stanovení je potřeba minimálně osmi dvojic odpovídajících si bodů.
- RANSAC – Slovo RANSAC je zkratkou slovního spojení RANdom Sample Consensus(doslovně přeloženo jako „Náhodná shoda vzorku“). Jedná se o robustní metodu. Z matematicky historického hlediska se jedná o poměrně mladou metodu. Její algoritmus byl poprvé publikován

v roce 1981 dvojicí Robert C. Bolles a Martin A. Fischler. Je to metoda, vybírající náhodnou skupinu dat ze všech měření a je určena pro odhad parametrů matematického modelu ze souboru sledovaných dat. Předpokládáme, že naměřená data jsou složena ze vzorků, odpovídajících měřenému modelu a ze vzorků, odpovídajících měřícím chybám. Chyby můžou být způsobeny šumem, chybným měřením, nebo chybnou interpretací dat. Při této metodě se automaticky počítá s vyřazením chybných vzorků a správným stanovením matematického modelu na základě souboru vybraných dat. Díky úplnému vyřazení chybných dat lze určit parametry modelu poměrně s vysokou přesností. Při výpočtu je vždy vybírán náhodný soubor dat. Z něj se vytvoří model, vůči němuž jsou poté testovány další vzorky. Jestliže vzorek přibližně odpovídá modelu, je vyhodnocen jako bezchybný. V opačném případě je vzorek vyřazen. Tento postup se několikrát opakuje až do získání nejlepšího modelu s nejmenší chybou. Počet opakování závisí na chybovosti vzorků a není předem znám. Tato metoda je pravděpodobně i s metodou LMedS nejvhodnější variantou výpočtu fundamentální matice při automatické detekci korespondujících bodů ve stereo-páru kvůli vyřazení chybně určených korespondencí. Kvůli vyřazování části vzorků nelze jednoznačně určit nejnižší potřebný počet korespondujících bodů.

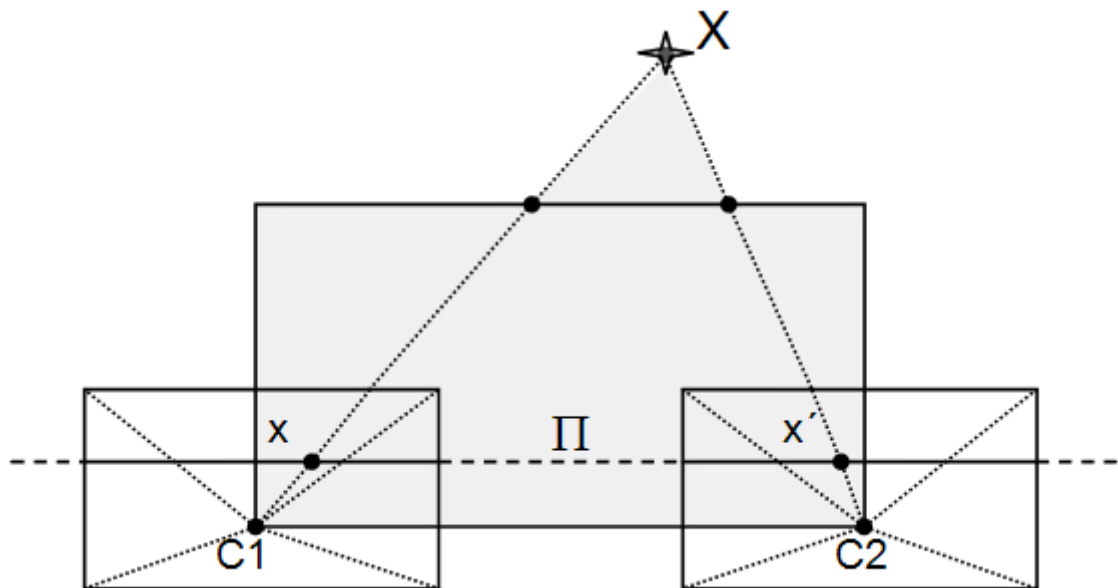
- LMedS – Je často užívaná zkratka slovního spojení Least Median Square, neboli metoda nejmenších čtverců, užívaná například při aproximaci matematických funkcí. Jedná se o robustní metodu. Podobně jako metoda RANSAC vyřazuje také nevyhovující body. Podle [4] může tolerovat až 50% nevyhovujících bodů tak, že až polovina vstupních bodů může být libovolně vzdálena od „pravdy“ bez objektivní změny funkce. Nelze-li medián určit jednoznačně, postupuje se podobně, jako u metody RANSAC. Namísto shody je každý vzorek hodnocen podle mediánu vstupních bodů. Nakonec je zvolen model s nejnižším mediánem.

2. JEDNODUCHÁ STEREOGEOMETRIE

Jednoduché stereogeometrie lze dosáhnout speciálním postavením kamer, popřípadě tzv. *rektifikací* obrazů. Jedná se o takové postavení, kdy jsou osy kamer rovnoběžné. V takovém případě epipolární přímky odpovídají řádkům obrázku a jsou rovnoběžné s osou obrázku.

2.1 REKTIFIKACE OBRÁZKŮ

Je proces, pomocí kterého se obrázek upraví tak, aby splňoval podmínku jednoduché stereo-geometrie a to aby byly epipolární přímky rovnoběžné s osou obrázku. Rektifikací stereo-páru mohou být jednotlivé obrázky považovány za zobrazení novými virtuálními kamerami, získanými rotací a změnou vnitřních parametrů kamer. Důležitou výhodou rektifikace je zjednodušení hledání stereo-korespondencí, neboť jsme oblast hledání eliminovali pouze na vertikální směr rektifikovaných obrázků.



Obr.5.: Jednoduchá stereogeometrie

3.MORPHING MEZILEHLÉHO POHLEDU

Podle [1]: Existují v podstatě dva rozdílné způsoby principy mezilehlého pohledu: warping a morphing. V případě warpingu se vytváří nový pohled deformací jednoho referenčního snímku na základě informace o disparitě. U morphingu uvažujeme tvorbu pohledu virtuální kamery jako plynulý přechod mezi dvěma rovnocennými snímky. V našem případě budeme tedy uvažovat morphing.

V případě obrázků, rektifikovaných do jednoduché stereo geometrie, tedy uvažujeme pouze vzájemné posunutí pohledů po ose x, bude mít vektor $\mathbf{t}$ disparitní rovnice tvar $\mathbf{t} = (-t_x, 0, 0)^T$ (platí pro směr od $\mathbf{C}'$ do $\mathbf{C}$). Jelikož rotační matice bude jednotková (nemá vliv na výsledek) a předpokládáme totožné vnitřní parametry kamer ($\mathbf{A}=\mathbf{A}'$), pak platí: $\mathbf{A}'\mathbf{R}\mathbf{A}^{-1}=\mathbf{A}'\mathbf{A}=\mathbf{I}$ ($\mathbf{I}$ značí jednotkovou matici). To v praxi znamená, že další zpracování obrazu již není ovlivněno ani vnitřními parametry kamer, ani vzájemnou rotací. Jelikož hloubková souřadnice Z obou bodů je totožná a body se liší pouze v x-ové souřadnici, pak je možné psát:

$$x_r = x_l - \frac{t_x \cdot \alpha_u}{Z};$$

α_u je vnitřní parametr kamery a t_x je vzdálenost projekčních středů kamer.

Uvažujme levý snímek, jako referenční a $L(x_l, y_l)$ jeho matice úrovní jasu jednotlivých pixelů. $R(x_2, y_2)$ je takovou maticí druhého snímku. Pro všechny pixely, kromě těch překrytých (okludovaných), lze v jednom rektifikovaném snímku nalézt jejich protějšky ve druhém snímku stereo-páru. Jestliže kvůli charakteru rektifikovaných snímků uvažujeme (v závislosti na hloubkové souřadnici Z) pouze posunutí v ose x ($y_l=y_2=y$) a označíme $d(x_l, y)$ jako dvourozměrné pole disparit d , pak lze napsat

$$x_2 = x_l - d(x_l, y)$$

Mezi zjištěnou disparitou a skutečnou hloubkovou souřadnicí lze také nalézt vztah:

$$d \cdot \text{šířka pixelu} = \frac{t_x \cdot \alpha_u}{Z}; \text{ šířka pixelu je pouze konstanta, pomocí níž}$$

se vztahuje jednotka disparity vzhledem k ostatním proměnným. V podstatě

nám ale stačí, že disparita s hloubkovou souřadnicí jsou na sobě lineárně závislé (hloubka v reálné scéně stoupá, disparita klesá a naopak).

Známe-li hodnotu disparity, můžeme potom souřadnice mezilehlého pohledu (x_m, y_m) vypočítat takto:

$$x_m = x_l - d \cdot \alpha; \text{ a } y_m = y$$

α je transformační koeficient, nabývající hodnot $<0, 1>$ a definuje polohu virtuální kamery (0 – úplně vlevo, 1 – úplně vpravo)

4. POČÍTAČOVÉ ZPRACOVÁNÍ STEREO-PÁRU

V zásadě existují podle [3] dva přístupy k rekonstrukci prostorové scény. První možností je určení projekčních matic kamer, tj. získání vzájemných vztahů promítacích paprsků a pixelových souřadnic snímků vzhledem k pevně dané soustavě souřadnic. Tento přístup ovšem vyžaduje znalost vnitřních parametrů kamery, v opačném případě je třeba odhadnout pro každý snímek 11 parametrů projekční matice. Vnitřní parametry kamery se zjišťují tzv. kalibrací kamery. Kalibrační techniky je možné rozdělit na fotogrammetrickou kalibraci a automatickou kalibraci. Při fotogrammetrické kalibraci pozorujeme kamerou kalibrační objekt, jehož geometrie je přesně určena. Výhodou je přesnost výsledků, avšak za cenu nákladného laboratorního vybavení. Při automatické kalibraci nepotřebujeme kalibrační objekt, vnitřní parametry kamery odhadneme na základě vzájemně si odpovídajících bodů pozorovaného objektu.

Vzhledem k zadání je práce zaměřena na druhou možnost, a to automatickou kalibraci, avšak pouze na základě ručně zadaných korespondencí pomocí fundamentální matice. Pro výpočet fundamentální matice byly využity již předdefinované funkce knihovny OpenCV od firmy Intel. Jedná se o rozsáhlý soubor open-source knihoven pro počítačové vidění. Obsahuje velké množství funkcí a procedur, využitelných při obrazových aplikacích. Knihovna obsahuje právě již zmíněné nejpoužívanější algoritmy pro výpočet fundamentální matice, a to algoritmy: 7-bodový, 8-bodový, RANSAC a LMedS.

Při prozkoumání syntaxe je následný výpočet velice jednoduchý. Vždy je potřeba vstupní soubor korespondujících bodů. Jeho získání může být provedeno různými způsoby. Automatická volba je možná například pomocí Harrisova detektoru rohů, MSER(Maximally Stable Extremal Regions) detektoru, algoritmu SIFT a dalších. Automatické metody ovšem produkují relativně velké množství bodů a chyb, tedy použití lineárních, nerobustních algoritmů výpočtu fundamentální matice by mohlo vést k chybným výsledkům. Podle zadání byla práce zaměřena nejprve na ruční zadávání korespondujících bodů, v kapitole 4.2.2. je však popsán i jeden s algoritmů automatického rozpoznávání odpovídajících si bodů.

Nyní zde popíši fungování přiloženého programu s popisem některých funkcí, zejména knihovny OpenCV:

4.1.NAČTENÍ STEREO-PÁRU DO PAMĚTI A PŘÍPRAVA PRO PRÁCI

Nejdříve je potřeba stereo-pár nějakým způsobem získat (myšleno např. vyfotografovat, přehrát do počítače) a poté provést:

4.1.1: Načtení do paměti

Pro práci s obrázky knihovna OpenCV obsahuje datový typ `IplImage`.

Načtení probíhá pomocí procedury `cvLoadImage` s následující syntaxí:

```
IplImage* img=cvLoadImage("img.xx");
```

„*img*“ je nová proměnná typu `IplImage`, „*img.xx*“ bude nahrazeno konkrétní cestou k souboru.

4.1.2 Vytvoření okna

Je potřeba vytvořit nové okno pro zobrazení obrázku. K tomu slouží procedura `cvNamedWindow` se syntaxí: `cvNamedWindow("window", 0);`

„*window*“ bude identifikátor nového okna a zároveň se tento nápis zobrazí v hlavičce okna, „0“ je volitelný parametr, který může nabývat dvou hodnot:

- `CV_WINDOW_AUTOSIZE` – automatická velikost okna, přičemž velikost se nedá upravovat.
- `0` – velikost okna podle nastavení při posledním spuštění, velikost okna se dá takto upravovat, přičemž původní poměr stran nezůstává zachován.

4.1.3 Zobrazení obrázků

Samotné zobrazení obrázků do okna již proběhne pomocí procedury `cvShowImage` následovně:

```
cvShowImage("window",img);
```

window je identifikátor okna, které bylo pro tento účel vytvořeno, *img* je název proměnné, do níž byl načtený daný obrázek.

Tyto tři body provedeme pro oba obrázky stereo-páru a můžeme již přikročit k samotnému zadávání korespondencí.

4.2. ZADÁVÁNÍ KORESPONDENCÍ:

4.2.1 Ruční zadávání

V případě ručního zadávání korespondujících bodů je pravděpodobně nejjednodušší přímé označení korespondencí kurzorem myši. Toto zadávání samozřejmě bude také zatíženo chybou danou zejména faktory jako rozlišení obrazovky, zvolené přiblížení, přesnost myši, nebo také kvalita zraku uživatele, preciznost při volbě a schopnost práce s myší.

Každý počítačový obraz je vlastně matice bodů o velikosti $X \times Y$, kde X je horizontální rozlišení a Y vertikální. Při hledání konkrétního bodu na obrázku hledáme de facto prvek dané matice, který lze najít pomocí souřadného systému.

Knihovna OpenCV obsahuje nástroj pro zjištění činnosti, vykonávané v daném okně, vracející aktuální souřadnice kurzoru a vykonanou činnost. Jedná se o tzv. „callback“ funkci, kterou je nutné inicializovat na začátku běhu programu, v hlavní funkci, nebo dříve, než bude nutné ji využívat.

Inicializace probíhá pomocí funkce `cvSetMouseCallback` se syntaxí:

```
cvSetMouseCallback( "window" , on_mouse);
```

`window` je identifikátor okna, ve kterém když vykonáme určitou činnost myši, zavolá se funkce `on_mouse`, která musí být deklarována ve formátu:

```
void on_mouse(int event, int x, int y, int flag, void* param)
```

Vstupní proměnné „*event*“ mohou být přiřazeny hodnoty:

`CV_EVENT_MOUSEMOVE` (pohyb myši),
`CV_EVENT_LBUTTONDOWN` (levého tlačítka myši je stlačeno),
`CV_EVENT_RBUTTONDOWN` (pravé tlačítka myši je stlačeno),
`CV_EVENT_MBUTTONDOWN` (prostřední tlačítka myši je stlačeno),
`CV_EVENT_LBUTTONUP` (levé tlačítka myši není stlačeno),
`CV_EVENT_RBUTTONUP` (pravé tlačítka myši není stlačeno),
`CV_EVENT_MBUTTONUP` (prostřední tlačítka myši není stlačeno),
`CV_EVENT_LBUTTONDBLCLK` (dvojklik levým tlačítkem myši),
`CV_EVENT_RBUTTONDBLCLK` (dvojklik pravým tlačítkem myši),
a `CV_EVENT_MBUTTONDBLCLK` (dvojklik prostředním tlačítkem myši).

Proměnným x a y jsou přiřazeny aktuální x-ové a y-ové souřadnice, `flag` je další příznak prováděné akce(`CV_EVENT_FLAG_LBUTTON`, `CV_EVENT_FLAG_RBUTTON`, `CV_EVENT_FLAG_MBUTTON`, `CV_EVENT_FLAG_CTRLKEY`, `CV_EVENT_FLAG_SHIFTKEY`, `CV_EVENT_FLAG_ALTKEY`), `param` je volitelný parametr.

Pro další použití si musíme uvědomit, co po funkci `on_mouse` vlastně požadujeme a tento požadavek realizovat například pomocí podmínky. V našem případě potřebujeme získat souřadnice okna při kliknutí levým tlačítkem do obrazu a načíst je do proměnné typu `CvMat` (matice), jelikož zamýšlíme tyto body použít k výpočtu fundamentální matice. Přístup k prvkům dvousloupcové matice se realizuje pomocí

`mat->data.fl[q]=x;` kde `mat` je název proměnné, `fl` je datový typ (zde `float`, může být i `i=integer`, `db=double`, `s=short`, `ptr=unsigned char`), `q` je číslo prvku. Sudé prvky značí první sloupec (x-ové souřadnice), liché druhý (y-ové souřadnice).

Získání souřadnic bodu tedy proběhne v rámci funkce „`on_mouse`“ takto:

```
if( event == CV_EVENT_LBUTTONDOWN ) {mat->data.fl[i*2]=x;
mat->data.fl[i*2+1]=y; }
```

`i` zde značí číslo řádku, tedy `i` pořadové číslo označeného bodu. Nyní již je možné pomocí vhodné procedury načíst do paměti všechny potřebné korespondence v obou obrázcích.

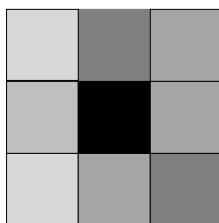
4.2.2 Automatické hledání korespondencí pomocí algoritmu SIFT

Podle [9]: SIFT, neboli Scale-Invariant Feature Transformation je algoritmus, který na základě jasových informací podle zadaných parametrů vytváří v obrazech tzv. body zájmu, ze kterých pak dalším porovnáváním získává páry korespondujících bodů. Scale-Invariant znamená, že rozlišovací schopnost tohoto algoritmu je nezávislá na použitém měřítku či rotaci. Je také velice odolný vůči změnám v osvětlení scény, obrazovému šumu a mírným změnám úhlu pohledu. Právě odolnost vůči změně úhlu pohledu je důvodem použití právě v naší aplikaci.

Postup:

1. Nejprve je potřeba najít v obrazech body zájmu, tzn. takové body, v jejichž okolí dochází k vysokým změnám v jasu. To se provede několikanásobným gaussovským rozostřením a následně vzájemným odečtením obou rozostřených obrázků. Čím vícekrát rozostření provedeme, tím více bodů můžeme získat ovšem za cenu délky výpočetního času. Autorem algoritmu byl počet těchto kroků experimentálně určen na 3. Tento postup provedeme pro různá měřítka obrazu.

2. Po odečtu rozostřených obrazů v nich hledáme lokální extrém, tedy pixely s hodnotou jasu výrazně rozdílnou od sousedů. Hledání probíhá v osmiokolí pixelu, tedy v matici s rozměrem 3x3 ve všech měřítkách.



Obr.6.: Hledání lokálních extrémů

Odfiltrujeme nestabilní body zájmu, tedy málo kontrastní, na hranách a na velkých homogenních plochách.

3. Pro dané body zájmu vypočítáme hodnoty tzv. deskriptorů, typicky 128 bitových vektorů, vypočítaných podle gradientů okolních bodů. Ty obsahují informaci o velikosti ($L = \sqrt{d_x^2 + d_y^2}$) a orientaci ($\alpha = \arctan(dy/dx)$).

4. Pro samotné získání páru korespondencí porovnáváme deskriptory bodů zájmu mezi jednotlivými snímky. Při dostatečné míře podobnosti jsou pak tyto body zájmu vyhodnoceny jako korespondence. Z těchto korespondencí je potřeba ještě vyloučit všechny duplicitní, nulové a korespondence mimo obraz. Zbývající jsou poté uloženy do matice a jsou připraveny k dalšímu použití. V našem případě pro výpočet fundamentální matice stereo-páru.

4.3. VÝPOČET FUNDAMENTÁLNÍ MATICE

Samotný výpočet fundamentální matice je již pomocí funkcí OpenCV již velmi jednoduchý. Knihovna obsahuje 4 algoritmy pro výpočet fundamentální matice. K výpočtu se používá funkce `cvFindFundamentalMat`, která vrátí počet určených matic (0 když nemohla být stanovena a 1 byla-li stanovena, popř. až 3 u 7-bodového algoritmu) a samotnou vypočtenou matici, přičemž algoritmus se volí pomocí parametru. Syntaxe volání funkce je následující:

```
fm_pocet=cvFindFundamentalMat(mat1,mat2,fundamental_matrix,  
param_algo,param1,param2,status );
```

`fm_pocet` je proměnná typu integer, do které se uloží počet funkcí stanovených fundamentálních matic.

`mat1, mat2` jsou vstupní matice korespondencí v obou obrázcích.

`fundamental_matrix` je proměnná typu `CvMat`, do níž se uloží výstupní fundamentální matice.

`param_algo` je parametr volby algoritmu a může mít hodnoty:

`CV_FM_7POINT` pro výpočet pomocí sedmibodového algoritmu (počet vstupních korespondencí musí být přesně 7),

`CV_FM_8POINT` pro osmibodový algoritmus (počet vstupních korespondencí musí být 8 a více),

`CV_FM_RANSAC` pro algoritmus RANSAC (počet vstupních korespondencí musí být vyšší, než 8) a

`CV_FM_LMEDS` pro metodu nejmenších čtverců (počet vstupních korespondencí musí být vyšší, než 8).

`param1, param2` a `status` se zadává pouze u metod RANSAC a LMedS a jejich význam je

`param1` - největší vzdálenost mezi epipolární přímkou a příslušným bodem, může mít hodnotu 1-3.

`param2` - požadovaná jistota správnosti odhadu fundamentální matice, hodnota 0-1;

`status` - volitelné pole o počtu prvků, jako počet korespondujících bodů, každému bodu je přiřazena 0, jestliže nevyhovuje dané epipolární geometrii, nebo 1 jestliže vyhovuje a bude při výpočtu brán v úvahu.

4.4. ROZDĚLENÍ OBRÁZKŮ NA ŘÁDKY:

Tento proces je potřeba vzhledem k dalšímu postupu. Provádí se pomocí příkazu `cvMakeScanlines` se syntaxí:

```
cvMakeScanlines(    &FMAT,    cvSize(img->width,img->height),  
scanlines1, scanlines2, delkyL, delkyR, &line_count);
```

`FMAT` je fundamentální matice převedená do datového typu `CvMatrix3` (je zapotřebí konverze), v závorce `cvSize` jsou rozměry obrázků, `scanlines1` a `scanlines2` jsou podle [6] ukazatele na pole vypočítaných řádků, `delkyL` a `delkyR` jsou podle [6] ukazatele na pole vypočítaných délek řádku v pixelech, `line_count` uchovává počet řádků.

4.5 REKTIFIKACE OBRÁZKŮ

Provádí se pomocí funkce `cvPreWarpImage` se syntaxí

```
cvPreWarpImage(line_count,img1,buffer_1,delkyL,scanlines1);
```

Provádí se pro každý obrázek zvlášť – zde příklad pro první obrázek. `buffer_1` je prostor pro uložení dat rektifikace obrázku a musí být pro něj vyhrazeno místo v paměti ještě před voláním funkce.

Nyní již máme oba obrázky v paměti rektifikovány do modelu jednoduché stereo-geometrie a můžeme přikročit k dalším procedurám k vytváření mezilehlého pohledu.

4.6 ROZDĚLENÍ ŘÁDKŮ NA BĚHY

Touto procedurou již začíná hledání korespondencí mezi rektifikovanými obrázky, dojde k rozdělení jednotlivých řádků rektifikovaných obrázků na několik tzv. „běhů“, to znamená na soubory pixelů s podobným jasem. Provádí se za pomoci funkce `cvFindRuns` podle syntaxe

```
cvFindRuns(line_count,buffer_1,buffer_2,delkyL,delkyR,  
runs1,runs2,num_runs1,num_runs2);
```

`runs1` a `runs2` jsou pole typu `integer` pro uchování souřadnic řádků v obrázcích.

`num_runs1` a `num_runs2` jsou pole typu `integer`, kde je uložen počet běhů pro každý řádek.

4.7 HLEDÁNÍ KORESPONDENCÍ MEZI BĚHY

V dalším kroku musíme nalézt korespondence mezi „běhy“ jednotlivých rektifikovaných obrázků. Toto je možné provést za pomoci funkce

`cvDynamicCorrespondMulti` se syntaxí

```
cvDynamicCorrespondMulti(line_count, runs1, num_runs1, runs2,  
num_runs2, first_cor, second_cor);
```

`first_cor` a `second_cor` jsou celočíselná pole, nesoucí právě informace o korespondencích, nalezených v jednotlivých bězích.

4.8 VYTVOŘENÍ ŘÁDKOVÁNÍ VIRTUÁLNÍ KAMERY

V tomto kroku nalezneme souřadnice řádků pohledu z virtuální kamery v dané pozici (podle váhového parametru `alpha`). Syntaxe pro tento úkon zní

```
cvMakeAlphaScanlines(scanlines1, scanlines2, scanlines_p,  
delky, line_count, alpha);
```

`scanlines_p` je ukazatel na pole řádků, vytvořených pro virtuální kameru.

`alpha` je váhový parametr typu `float`, nabývající hodnot 0-1 a určuje polohu virtuální kamery mezi reálnými snímky.

4.9 MORPHING OBRÁZKŮ

Při použití funkce `cvMorphEpilinesMulti` dojde k morphingu rektifikovaných obrázků za použití informací o korespondencích mezi řádky jednotlivých obrázků. Syntaxe zadání je následující:

```
cvMorphEpilinesMulti(line_count, buffer_1, delkyL, buffer_2,  
delkyR, buffer_c, delky, alpha, runs1, num_runs1, runs2,  
num_runs2, first_cor, second_cor);
```

`buffer_c` má podobnou funkci jako `buffer_1` ve funkci `cvPreWarpImage`, je to dočasný zásobník pro ukládání výsledků výpočtů dané funkce.

4.10 VYTVOŘENÍ VÝSLEDNÉHO POHLEDU Z VIRTUÁLNÍ KAMERY

Pomocí funkce `cvPostWarpImage` se provede zpětné zakřivení výsledného obrazu tak, že klade pixely původních obrázků na řádky nalezené funkcí `cvMakeAlphaScanlines`.

```
cvPostWarpImage(line_count,    buffer_c,    delky,    stereo,  
scanlines_p);
```

Jediný nový parametr `stereo` značí nový obrázek.

4.11 VÝSLEDNÉ VYHLAZENÍ

V nově vytvořeném obrázku se mohou vyskytovat prázdná (tmavá) místa kvůli možným mezerám mezi sousedními řádky. Za pomoci funkce `cvDeleteMoire` dojde k odstranění takových míst nahrazením hodnoty tmavého pixelu hodnotou vytvořenou zprůměrováním okolních pixelů. Syntaxe je už velmi jednoduchá. Jediný parametr (`stereo`) značí obrázek, reprezentující výsledný pohled.

```
cvDeleteMoire(stereo);
```

Nyní již můžeme zobrazit nově vytvořený obrázek mezilehlého pohledu.

5.STRUČNÝ POPIS PROGRAMU StereoG:

V rámci této bakalářské práce jsem vytvořil jednoduchou aplikaci, která demonstruje zde popsané pojmy.

Implicitně po spuštění vidíme po levé straně ovládací panel a dva referenční obrázky stereo-páru:

Na panelech „Souřadnice“ se zobrazují aktuální souřadnice kurzoru myši v daném obrázku.



Obr.7.: Vzhled programu StereoG

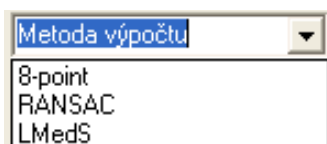
Při kliknutí na bod v obrázku se jeho souřadnice uloží do proměnné typu CvMat, příslušné pro daný obrázek a kolem bodu se vytvoří kružnice. Máme-li již označené korespondující body v obou obrázcích, je nutné kliknout na tlačítko „Další pár“. Tím dojde zjednodušeně ke změně aktuálního řádku paměti proměnných CvMat a je možné zadávat souřadnice dalšího páru

korespondujících bodů. Zároveň se přidají položky do seznamu „Odpovídající si body“ včetně pořadového čísla páru. Párů je potřeba vzhledem k použitým algoritmům potřeba zadat alespoň 8, nicméně doporučuji jich zadat alespoň 11 kvůli charakteristice výpočtů.

Do programu bylo také implementováno automatické rozpoznávání korespondencí pomocí algoritmu SIFT, jenž byl naprogramován Robertem Hessem a je šířen pod licencí GNU GPL. Bylo však nutné upravit vnitřní strukturu některých implementovaných funkcí. Tento algoritmus byl vytvořen a v roce 1999 publikován a patentován Davidem Lowem, kanadským vědcem a profesorem na univerzitě v Britské Kolumbii a je použitím vnitřních funkcí založen také na bázi OpenCV. Princip tohoto algoritmu je popsán výše. Samotné automatické rozpoznání program provede na základě stisknutí příslušného tlačítka na ovládacím panelu. Tyto korespondence budou také vyznačeny v jednotlivých obrázcích a jejich souřadnice zaneseny do seznamu „Odpovídající si body“, samozřejmě budou i uloženy do příslušných matic.

I po provedení automatického rozpoznání je možné individuálně přidávat další body ručně.

Jestliže máme již všechny páry zadány, po rozbalení nabídky „Metoda výpočtu“, máme na výběr ze tří metod výpočtu fundamentální matice, a to 8-bodový algoritmus, RANSAC a LMedS.



Po výběru algoritmu kliknutím na tlačítko „Vytvoř fundamentální matici“ se podle vybraného algoritmu zobrazí tabulka 3x3, v níž bude vepsána vypočtená fundamentální matice stereo-páru. Jestliže pomocí daného algoritmu nemohla být matice vytvořena, zobrazí se v tabulce chybové hlášení „NELZE VYTVOŘIT FUNDAMENTÁLNÍ MATICI!!!!“. Zároveň se v paměti provede na základě vypočtené fundamentální matice rozklad obrázků na řádky a je provedena i rektifikace obou obrázků. I po vygenerování je možné přidávat další korespondující body, které jsou při dalším generování vzaty v úvahu. Při kliknutí na tlačítko „Vytvoř mezilehlý pohled“ bude na základě nastavení parametru alpha vygenerován pohled z virtuální kamery umístěné mezi vstupními snímky

a zobrazen v novém okně. Při změně parametru α pomocí slideru dojde k přepočítání výsledného obrázku. Výsledek je možné pomocí příslušného tlačítka uložit do adresáře s programem a bude pojmenován „stereo.jpg“.

Pouze pro zajímavost je implementováno i vypočtení a disparitní mapy pomocí Birchfield-Tomasiho metody při kliknutí na příslušné tlačítko.

6. VÝSLEDKY TVORBY MEZILEHLÉHO POHLEDU POMOCÍ APLIKACE StereoG A NEDOSTATKY



Alpha = 0 – Reálný pohled zleva



Alpha = 0,25 – virtuální kamera v $\frac{1}{4}$ dráhy



Alpha = 0,5 – virtuální kamera přesně mezi reálnými pohledy



Alpha = 0,75 – virtuální kamera ve $\frac{3}{4}$ dráhy



Alpha = 1 – reálný pohled zprava

Obr.8.: Výsledky tvorby mezilehlého pohledu pomocí aplikace StereoG

Nedostatky aplikace: Bohužel nebyl vyřešen problém s načítáním obrázků pomocí dialogového okna. Pro práci s jiným párem obrázků je třeba obrázky přejmenovat na jméno souborů původního páru (2L.jpg a 2R.jpg). Soubory se načítají z kořenového adresáře programu.

Je vidět, že v pohledech generovaných programem jsou patrné určité artefakty a mírné nepřesnosti ve vykreslení. To bylo dle mého názoru způsobeno zejména nemožností v některých místech vytvořit korespondence a naopak nemožností vyloučit některé chybně určené odpovídající si body. Při testování programu na různých snímcích se tento efekt jevil nejhorší, pokud se při automatickém rozpoznávání korespondencí v obraze vyskytovalo větší množství objektů podobného charakteru a algoritmus SIFT nemohl určit jednoznačně, který z bodů zájmu vyhodnotit jako tu správnou korespondenci. Konkrétně šlo o případy výskytu kostkové dlažby, tlačítka klávesnice a další. V takových případech je zřejmě nejjistější právě ruční forma zadávání korespondencí, která je ovšem podstatně pracnější a časově náročnější, navíc je také problematická přesnost zadání (záleží na obsluze, přesnosti myši, rozlišení monitoru atd.). Důsledkem těchto problémů by bylo chybné určení fundamentální matice a následná nepřesná rektifikace snímků.

Neméně závažná je ovšem i problematika zjišťování korespondencí mezi jednotlivými běhy (pomocí funkce `cvDynamicCorrespondMulti`). Mezi snímky nasnímaných z různých úhlů mohou být i mírné rozdíly v osvětlení scény, některé pixely se mohou překrývat a některé další faktory můžou způsobit, že dojde na základě chybně určené disparity k nekvalitnímu určení korespondencí mezi některými běhy.

Tyto faktory mají pravděpodobně největší vliv na výsledné zobrazení.

Aplikace, vytvořená pro tuto práci byla napsána v jazyku C++ pomocí prostředí Borland C++ Builderu 6.0 (dále jako zkratka BCB) s využitím souboru knihoven OpenCV. Kombinace OpenCV a BCB není úplně běžnou záležitostí a k implementaci není dostupný ucelený návod. Není to záležitost zcela triviální, proto byl v rámci této práce návod připraven.

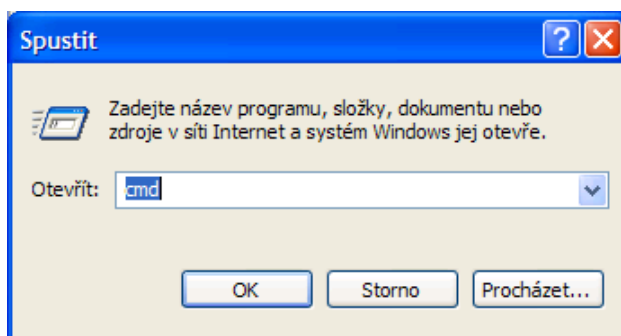
7.IMPLEMENTACE KNIHOVEN OPENCV DO BORLAND C++ BUILDERU

1.) Nejprve je třeba knihovnu stáhnout například ze stránek <http://sourceforge.net/projects/opencvlibrary/> (popřípadě dalších zdrojů, snadno naleznutelných na www.google.com) a nainstalovat.

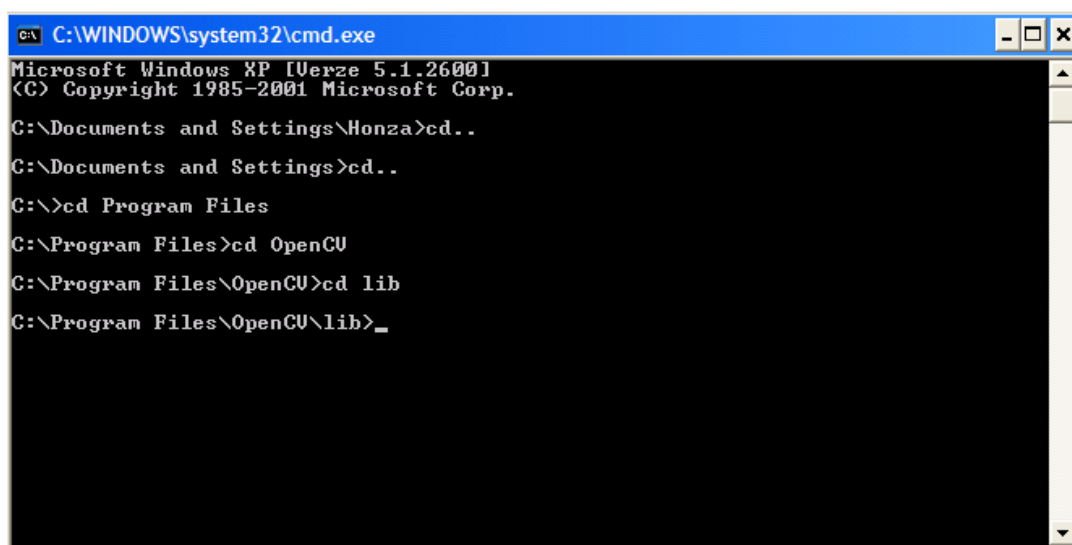
Pozn.: V tomto návodu budeme uvažovat výchozí adresář instalace, a to *C:\Program Files\OpenCV* . Při instalaci do jiného umístění bude samozřejmě nutná modifikace podle Vámi zvoleného adresáře.

2.) Jelikož BCB používá jiný a nekompatibilní formát(OMF)souborů *.lib, než používá OpenCV (COFF) je potřeba provést konverzi. Součástí BCB je nástroj pro konverzi s názvem COFF2OMF, obsluhovaný přes příkazovou řádku MS Windows, takže:

- Spustíme příkazovou řádku



- Pomocí příkazů systému MS-DOS *cd*(pro přechod do nižší úrovně) a *cd..* (přechod do vyšší úrovně) se dostaneme do adresáře *\OpenCV\lib*



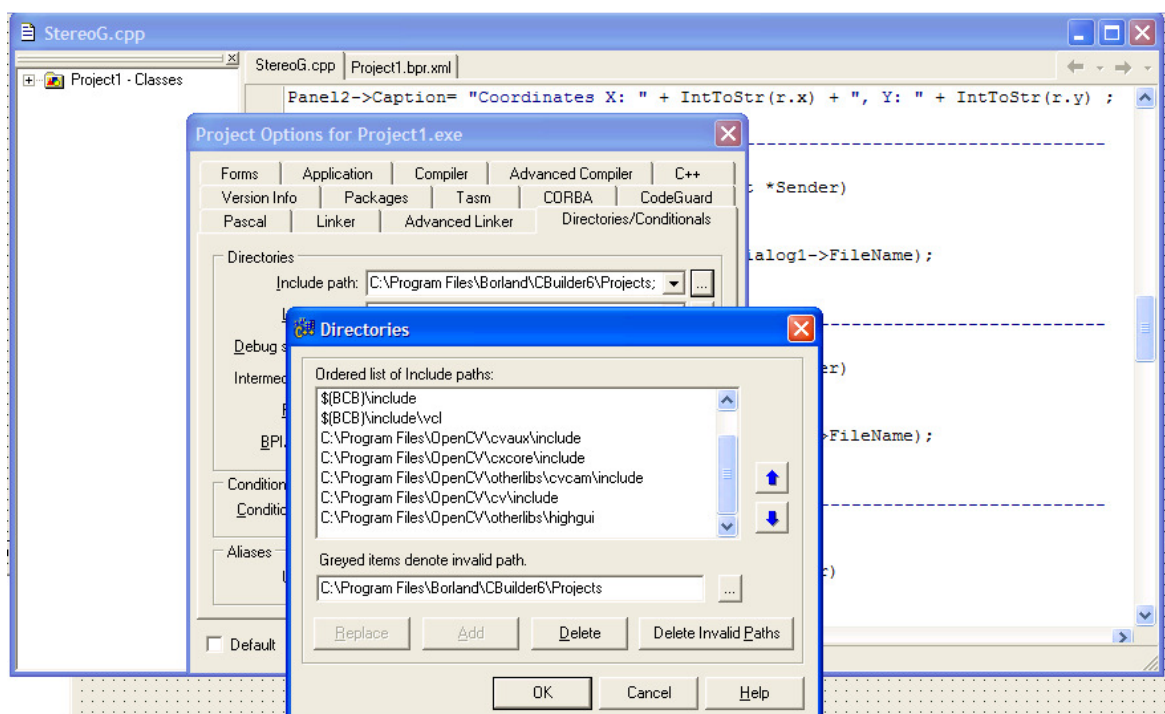
- Abychom si nepřemazali původní soubory, tak si příkazem „*mkdir borland*“ vytvoříme pracovní podadresář, v našem případě s názvem borland.
- V adresáři \OpenCV\lib se nachází 8 souborů: *cv.lib*, *cvaux.lib*, *cvcam.lib*, *cvhaartraining.lib*, *cxcore.lib*, *cxts.lib*, *highgui.lib* a *ml.lib*. Jelikož předem nevíme, jaké knihovny budeme potřebovat, provedeme konverzi všech a tím nám odpadnou případně potíže v budoucnosti.
- Samotná konverze se provádí příkazem `coff2omf` se syntaxí
„`coff2omf <puvodni_soubor> <novy_soubor>`“, např. :

```
C:\Program Files\OpenCV\lib>coff2omf cv.lib borland/cv.lib
COFF to OMF Converter Version 1.0.0.74 Copyright (c) 1999, 2000 Inprise Corporation
```

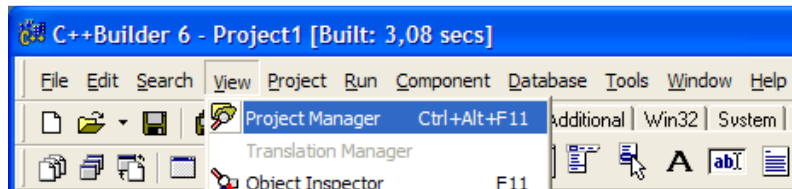
Úspěšný převod je potvrzen informací o programu.

3.) Jestliže jsme již provedli konverzi potřebných souborů, můžeme přikročit k samotné integraci knihovny do BCB.:

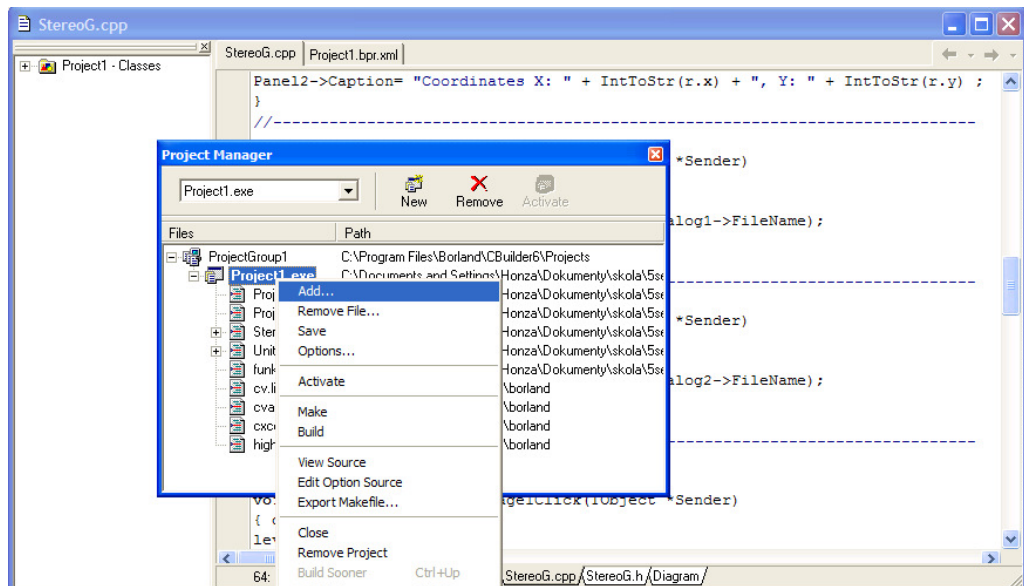
- Musíme do projektu zahrnout pracovní adresáře OpenCV se soubory, se kterými bude následně pracovat překladač. Klikneme na *Project->Options->záložka Directories/Conditionals->Include Path* a přidáme následující adresře: \OpenCV\cvaux\include , \OpenCV\cxcore\include, \OpenCV\otherlibs\cvcam\include, \OpenCV\cv\include, OpenCV\otherlibs\highgui



- Podobně jako v minulém bodě do seznamu Library path přidáme adresář se soubory, jejichž konverzi na formát omf jsme provedli(`\OpenCV\lib\borland`).
- Dále je nutné ještě přidat konvertované soubory to projektu takto: *View->Project Manager*



Klikneme pravým tlačítkem myši na náš projekt a zvolíme *Add*



Zde přidáme potřebné soubory z adresáře `\OpenCV\lib\borland`. Např. tyto základní:

	cv.lib	C:\Program Files\OpenCV\lib\borland
	cvaux.lib	C:\Program Files\OpenCV\lib\borland
	cxcore.lib	C:\Program Files\OpenCV\lib\borland
	highgui.lib	C:\Program Files\OpenCV\lib\borland

4.) Nakonec provedeme samotnou inkluzi celé knihovny přímo ve zdrojovém kódu projektu:

```
#include <cv.h>
#include <cxcore.h>
#include <cvaux.h>
#include <cvcam.h>
#include <highgui.h>
```

Nyní už můžete teoreticky využívat všech výhod této rozsáhlé knihovny počítačového vidění.

Na závěr lze jen dodat, že pravděpodobně pro všechny operace nebudete potřebovat všechny součásti knihovny OpenCV. Je to potřeba zvážit v rámci náročnosti překladu.

ZÁVĚR:

V rámci této bakalářské práce byla nastudována a popsána problematika epipolární geometrie. Byl zde naznačen i základní princip výpočtu fundamentální matice. V souvislosti s tím byl v práci popsán základní princip automatického hledání mezisnímkových korespondencí pomocí algoritmu SIFT. S využitím literatury byly také vysvětleny základy morphingu mezilehlého pohledu.

Podle zadání byla též naprogramována aplikace pro operační systém MS Windows, kterou jsem nazval StereoG. Pro tvorbu aplikace bylo využito vývojového prostředí Borland C++ Builder 6.0 (BCB) za pomoci knihoven OpenCV od firmy Intel. Jelikož není použití knihoven OpenCV v prostředí BCB zcela triviální, vytvořil jsem pro textovou část podrobný návod. Do programu StereoG byly implementovány tři algoritmy pro výpočet fundamentální matice stereo-páru za pomoci jak ručně zadaných korespondencí, tak automaticky získaných pomocí algoritmu SIFT, který byl s využitím funkcí od Roba Hesse pod licencí GNU GPL do programu také implementován. Program umožňuje také tvorbu mezilehlého pohledu z virtuální kamery, kterou je možno pomocí slideru umístit ve 100 krocích do libovolné polohy mezi reálnými pohledy.

Nepodařilo se mi bohužel zprovoznit načítání obrázků pomocí dialogového okna. Výsledné obrázky také obsahují určité nežádoucí artefakty plynoucí ze způsobu zpracování. Způsob eliminace těchto obrazových chyb by mohl být předmětem dalšího zkoumání.

Použitá literatura

- [1] Říha, K. *Pokročilé techniky zpracování obrazu*. Skripta FEKT VUT, 2007.
- [2] Hartley, R. I., Zisserman, A. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004
- [3] Voráčová, Š. 2004. *Aplikace epipolární geometrie*. Sborník 24. konference Geometrie a počítačové grafiky, Barborka, 2004, dostupné na: <http://www.fd.cvut.cz/personal/voracova/epipolar.pdf>
- [4] Fisher R.: *CVonline: The Evolving, Distributed, Non-Proprietary, On-Line Compendium of Computer Vision*. School of Informatics University of Edinburgh, dostupné na: <http://homepages.inf.ed.ac.uk/rbf/CVonline/>, [cit. 2008-11-28]
- [5] Xavier Armangué, Joaquim Salvi: *Overall view regarding fundamental matrix estimation*, Image and Vision Computing 21, pp-205-220, 2003..
- [6] Referenční manuál CvAux, dostupný na internetových stránkách: <http://opencv.willowgarage.com/wiki/CvAux>, [cit. 2009-05-20]
- [7] Bradski, D. G., Kaehler, A. *Learning OpenCV, 1st Edition*. First. O'Reilly Media, Inc., 2008
- [8] Brunet, T., Chao, L., *Uncalibrated Stereo Vision*, A CS 766 Project University of Wisconsin – Madison, 2004, dostupné na internetových stránkách: <http://pages.cs.wisc.edu/~chaol/cs766/cs766.html>, [cit. 2009-05-20]
- [9] HESS, Rob. *Rob Hess School of EECS @ Oregon State University* [online]. 2007. Dostupné na: <http://web.engr.oregonstate.edu/~hess/index.html> [cit. 2009-05-20]

Poznámka: literatura [7] nebyla v textu přímo ocitována, nicméně mi výrazně pomohla k pochopení některých souvislostí. Proto je v seznamu uvedena.

Popis přílohy:

Součástí práce je i přiložené CD, obsahující:

- Elektronickou verzi bakalářské práce
- Zdrojové kódy k aplikaci StereoG + projekt pro Borland C++ Builder 6.0
- Sadu testovacích stereo-párů

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma " **Úprava obecného stereo páru obrazu do jednoduché stereo geometrie**" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne

.....

(podpis autora)