

SEZMAM PŘÍLOH

Příloha 1: Zobrazení testované scény

Příloha 2: Spojený obraz pro dvoupohledový autostereoskopický displej

Příloha 3: Výsledné stereoskopické video

Příloha 4: Výsledné autostereoskopický snímek složený z 8 kamer

Příloha 5: Výsledné autostereoskopický snímek sestavený z 8 generovaných snímků

Příloha 6: Generovaný prostřední mezilehlý snímek

Příloha 7: optimalizace výpočtu složeného obrazu pro dvoupohledový autostereoskopický displej

Příloha 8: Ukázka zkráceného zdrojového kódu používající vytvořené DirectShow filtry

Příloha 9: Použití OpenCV funkcí pro generaci mezilehlého pohledu

Příloha 10: Konfigurační soubor pro generování obrazu pro vícepohledového displej

Příloha 11: Vliv optimalizace na rychlost výpočtu spojeného obrazu

Příloha 12: Hledání korespondencí na základě kalibračních obrazců

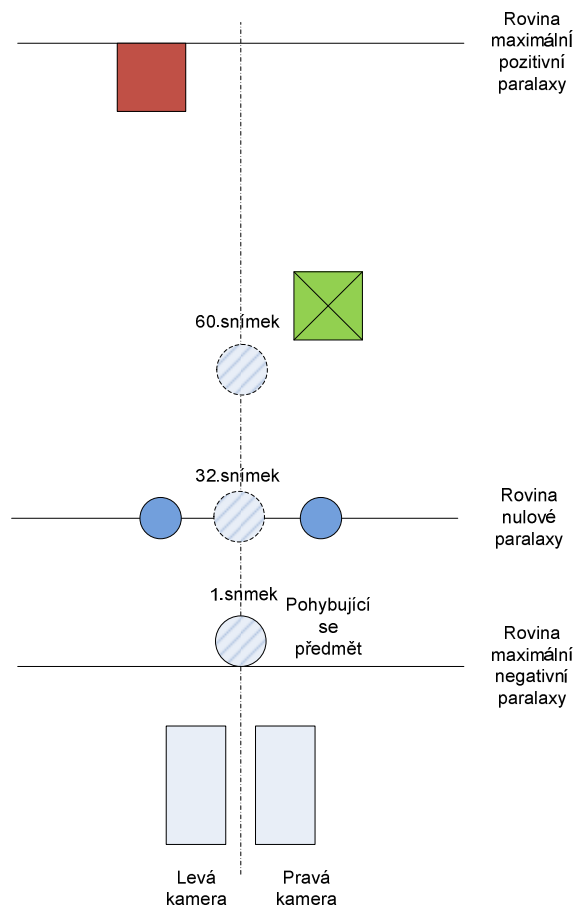
Příloha 13: Nekalibrovaná rektifikace

Příloha 14: Kalibrovaná rektifikace

Příloha 15: Obsah CD

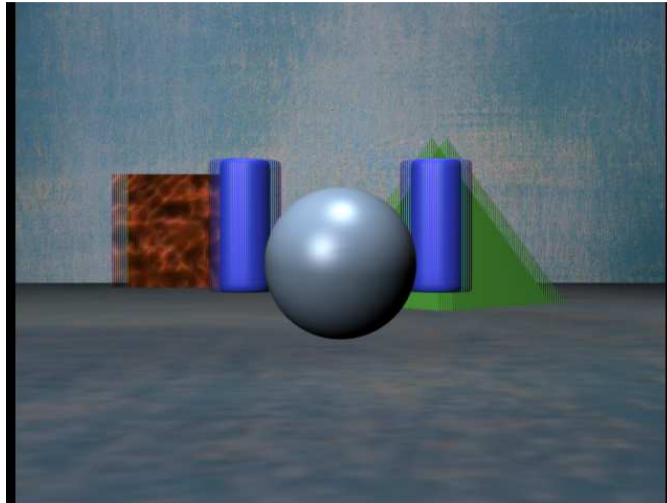
Příloha 16: CD

Příloha 1: Zobrazení testované scény



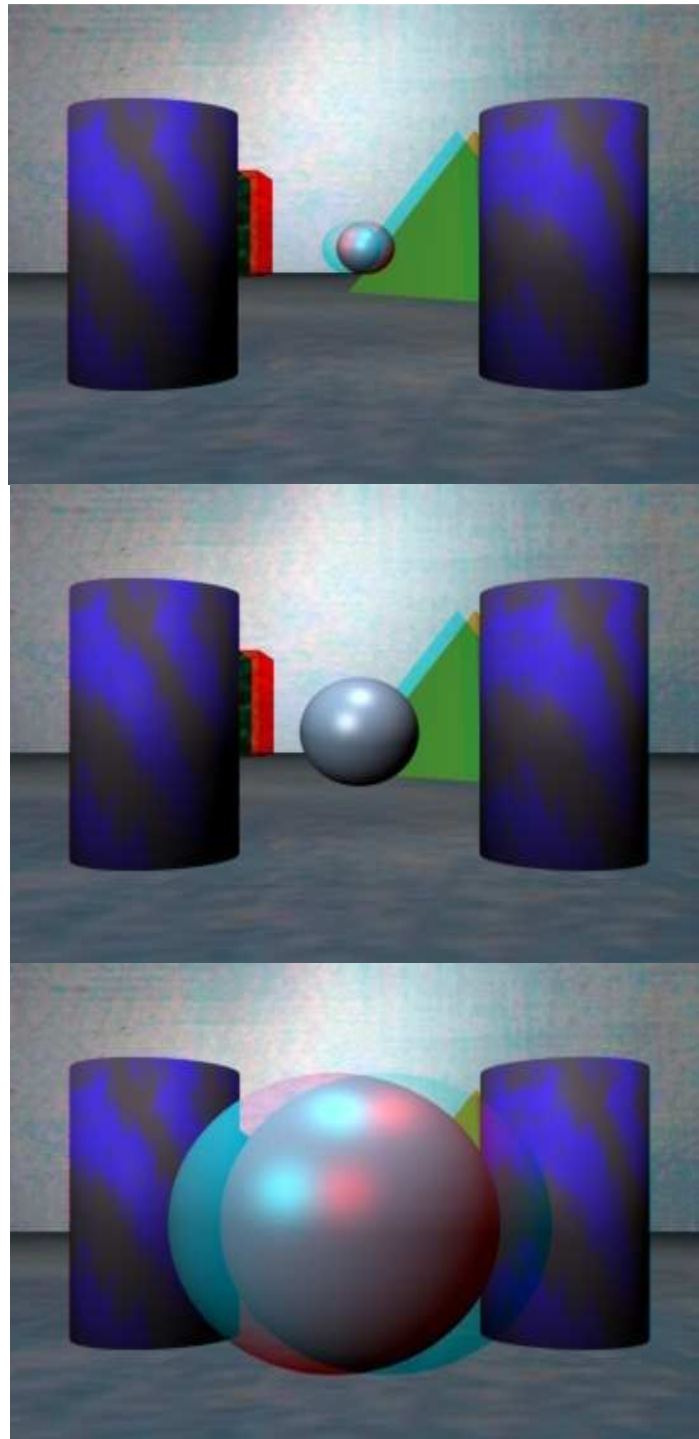
Nastavení scény: nejdálčenější bod 12m, hranice nulové paralaxy 2,5m, nejbližší bod 0.9m, zorný úhel kamery 50 stupňů a požadovaná šířka po ořezání 640 pixelů. Vypočtený rozestup kamer 89 mm pro maximální hodnotu paralaxy 3% z šířky výsledného snímku je přibližně 19 pixelů. Celá scéna vytvořená v Blenderu je k dispozici na přiloženém CD.

Příloha 2: Spojený obraz pro dvoupohledový autostereoskopický displej



Zobrazeno po posunutí a oříznutí snímku (**32. snímek**)

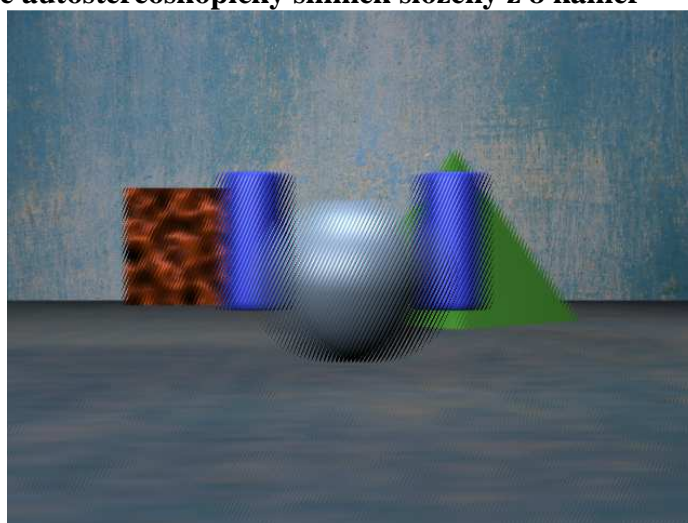
Příloha 3: Výsledné stereoskopické video



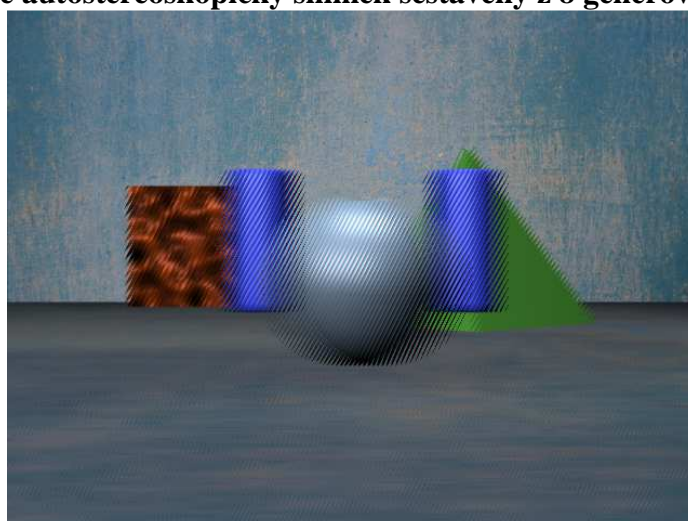
Ukázka získaná z druhé aplikace.

Ukázka zespoda: 1. snímek, 32. snímek, 60. snímek.

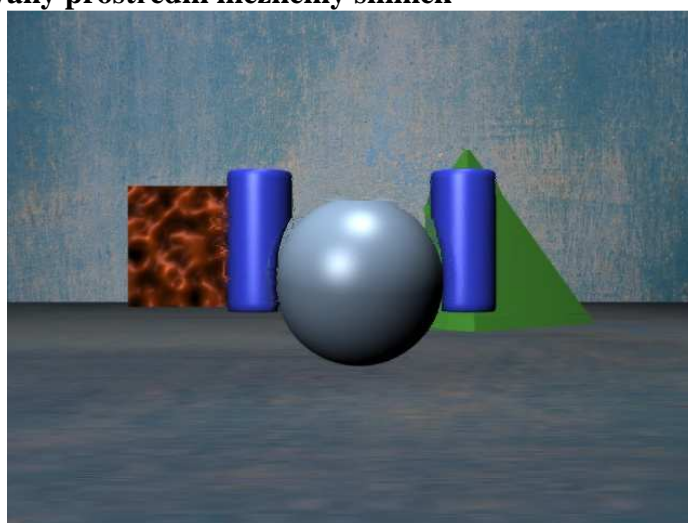
Příloha 4: Výsledné autostereoskopický snímek složený z 8 kamer



Příloha 5: Výsledné autostereoskopický snímek sestavený z 8 generovaných snímků



Příloha 6: Generovaný prostřední mezilehlý snímek



Příloha 7: Optimalizace výpočtu složeného obrazu pro dvoupohledový autostereoskopický displej

```
inline void see_autostereoscopic( void *pLeft, void *pRight, int iLength ){
__declspec(align(16)) const short mask1[8] = {0xff,0,0xff,0xff,0,0xff,0,0};
__declspec(align(16)) const short mask2[8] = {0,0xff,0,0xff,0xff,0,0xff,0};

    __asm{
        mov     esi, [pLeft]           ; levý snímek
        mov     edi, [pRight]          ; pravý snímek
        mov     ecx, [iLength]         ; počet pixelu

        pxor    xmm2, xmm2             ; inicializace registru xmm2

        ; maska 1
        movdqa  xmm6, [mask1]          ; první maska
        packuswb xmm6, xmm6            ; zpakují se wordy do bytes

        ; maska 2
        movdqa  xmm5, [mask2]          ; druhá maska
        packuswb xmm5, xmm5

        start_loop:                    ; smyčka
            movdqu  xmm0, [esi]         ; načtení do xmm registru
            movdqu  xmm1, [edi]

            pand  xmm0, xmm6            ; maskovat L, a R slozky
            pand  xmm1, xmm5

            por   xmm0, xmm1           ; secti

            movdqu [esi], xmm0         ; uložení 4 pixelu(do pLeft)

            add    esi, 16              ; posun L o 16 bytu
            add    edi, 16              ; posun P o 16 bytu
            sub    ecx, 4               ; delka - 4pixely
            jnz   start_loop

        emms
    }
}
```

Ukázka kódu použitého v *Stereoscopic Transform* filtru (výsledek se ukládá přes levý snímek)

Příloha 8: Ukázka zkráceného zdrojového kódu používající vytvořené DirectShow filtry

```
HRESULT BuildGraphStereoOwnFuncCamera( int iCamLeft, int iCamRight,
PROPDATA _data, void (*pFunc)(IplImage *, IplImage *) ){

    ...

    // Zdroj kamer
    IBaseFilter * pCam1 = NULL, * pCam2 = NULL;
    hr = GetCaptureDevice( &pCam1, iCamLeft );
    hr = m_pGraph->AddFilter( pCam1, L"Camera Left" );

    hr = GetCaptureDevice( &pCam2, iCamRight );
    hr = m_pGraph->AddFilter( pCam2, L"Camera Right" );

    // Vytvoreni instance filtru „My Stereoscopic Muxer“
    IBaseFilter * pMux = NULL;
    hr = CoCreateInstance( CLSID_GUIMyMuxFilter, NULL, CLSCTX_ALL,
        IID_IBaseFilter, (void**)&pMux );
    hr = m_pGraph->AddFilter( pMux, L"Stereo MUX" );

    // nastaveni parametru „My Stereoscopic Muxer“ filtru
    IIPMyMuxFilter *pIMux = NULL;
    hr = pMux->QueryInterface( IID_IIPMyMuxFilter, (void**)&pIMux );
    hr = pIMux->put_IPEffect( ERL_LR ); // formal levy a pravy snimek
    SAFE_RELEASE(pIMux);

    // Vytvoreni instance filtru „Stereoscopic Transform“ filtru
    IBaseFilter * pTrf = NULL;
    hr = CoCreateInstance( CLSID_MyStereoTrf, NULL, CLSCTX_ALL,
        IID_IBaseFilter, (void**)&pTrf );
    hr = m_pGraph->AddFilter( pTrf, L"Stereo Transform filter" );

    // nastaveni parametru „My Stereoscopic Muxer“ filtru
    IIPMyStereoTrf *pITrf = NULL;
    hr = pTrf->QueryInterface( IID_IIPMyStereoTrf, (void**)&pITrf );
    hr = pITrf->put_IPEffect( m_propData ); // nahraje nastaveni
    hr = pITrf->put_CallBack( pFunc ); // pouziti vlastní funkce pro
        // zpracovani dat uvnitr filtru

    SAFE_RELEASE(pITrf);

    ...

    return hr;
}
```

Příloha 9: Použití OpenCV funkcí pro generaci mezilehlého pohledu

- 1.) `cvMakeScanlines`
Vypočítá souřadnice epipolárních přímků pro dva obrazy na základě fundamentální matice. Rozdělí obraz na řádky tzv. scanlines.
- 2.) `cvPreWarpImage`
Provedení rektifikace obrazu, epipolární přímky budou v horizontálním uspořádání
- 3.) `cvFindRuns`
funkce vyhledává scanlines z rektifikovaného obrazu a hledá sekvence pixelů ležících na epipolárních přímkách s přibližně stejným jasem
- 4.) `cvDynamicCorrespondMulti`
najde korespondence mezi dvěma sadami runs ze dvou obrazu
- 5.) `cvMakeAlphaScanlines`
Vypočítá souřadnice epipolárních přímků z virtuální kamery
- 6.) `cvMorphEpilinesMulti`
přetvoří dva rektifikované snímky použitím informací o stereokorespondenci
- 7.) `PostWarpImage`
Vytvoří výsledný mezilehlý obraz na základě epipolárních přímků vypočtených pro výsledný obraz
- 8.) `cvDeleteMoire`
Odstranění moire efektu na základě průměrování okolí pixelů

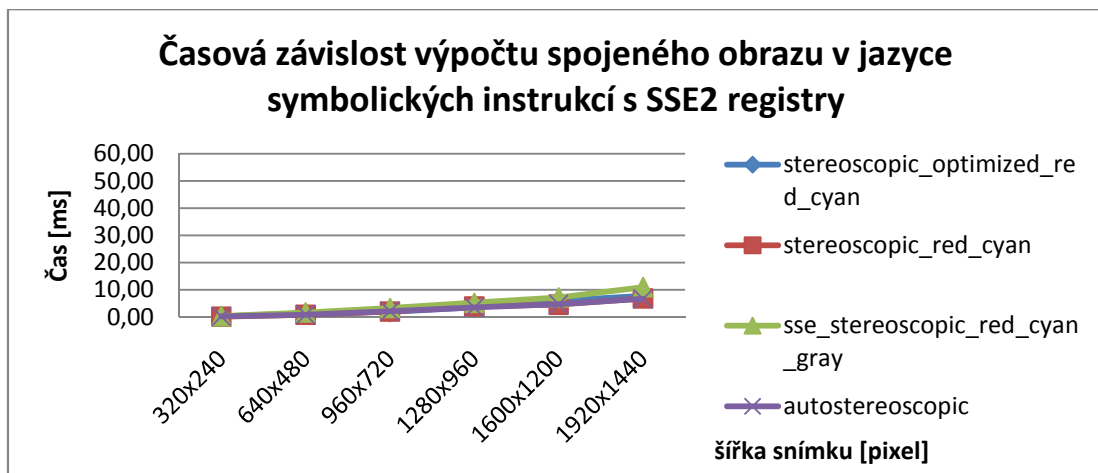
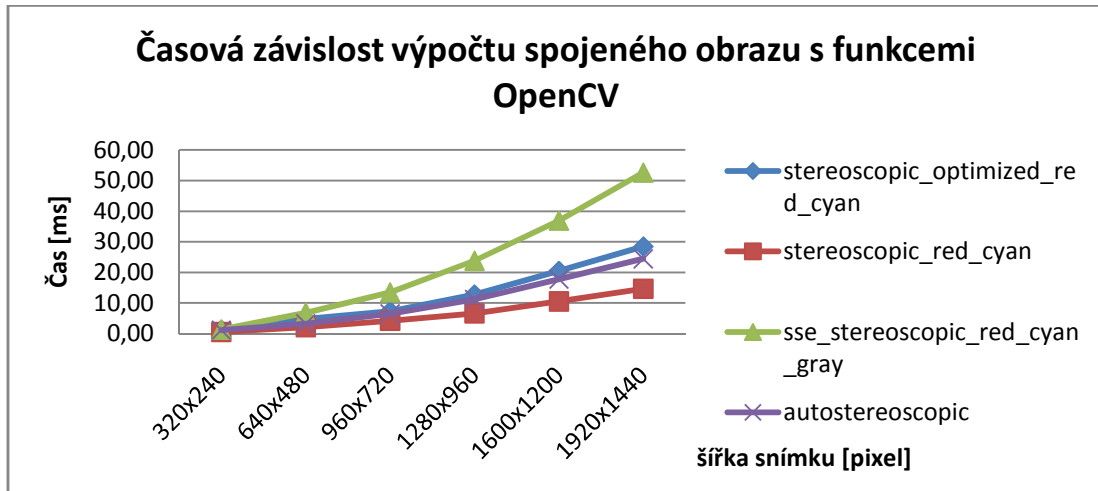
```
void Morphing( IplImage *result, double alpha );
```

Stručný popis funkce pro generování mezilehlého pohledu pro vícehledový autostereoskopický displej. Popsán je postup metody `Morphing(IplImage*, double)` ze třídy `CGenerateView`. Parametr `alpha` je v rozsahu od 0.0 do 1.0 a značí pozici mezilehlé kamery. Kde 0.0 je levý snímek a hodnota 1.0 je pravý snímek. Funkce je podrobně popsána ve zdrojovém kódu aplikace na CD.

Příloha 10: Konfigurační soubor pro generování obrazu pro vícehledového displej

```
# konfigurační soubor matice pro 8-pohledový autostereoskopický displej
# použití:
# sloupce, řádky;
# data oddělena ', ' ukončení řádku;
8,12;
0,1,2,3,4,5,6,7;
1,2,3,4,5,6,7,0;
1,2,3,4,5,6,7,0;
2,3,4,5,6,7,0,1;
3,4,5,6,7,0,1,2;
3,4,5,6,7,0,1,2;
4,5,6,7,0,1,2,3;
5,6,7,0,1,2,3,4;
5,6,7,0,1,2,3,4;
6,7,0,1,2,3,4,5;
7,0,1,2,3,4,5,6;
7,0,1,2,3,4,5,6;
```

Příloha 11: Vliv optimalizace na rychlost výpočtu spojeného obrazu

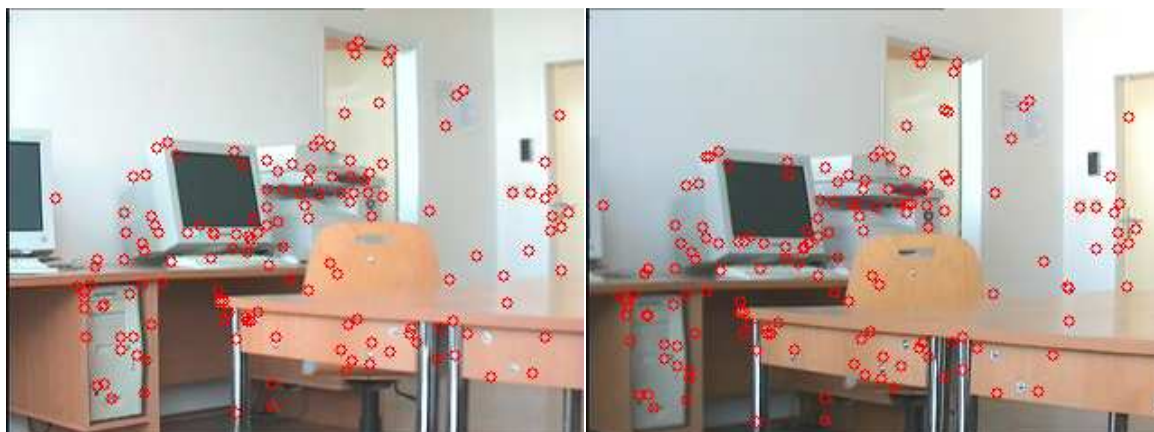


Vliv optimalizace použitím SSE2 registrů na rychlost výpočtu u vybraných metod. Testovaná sestava s procesorem Intel Core 2 Quad Q9400 2.6 GHz. Testovací aplikace i hodnoty jsou na přiloženém CD.

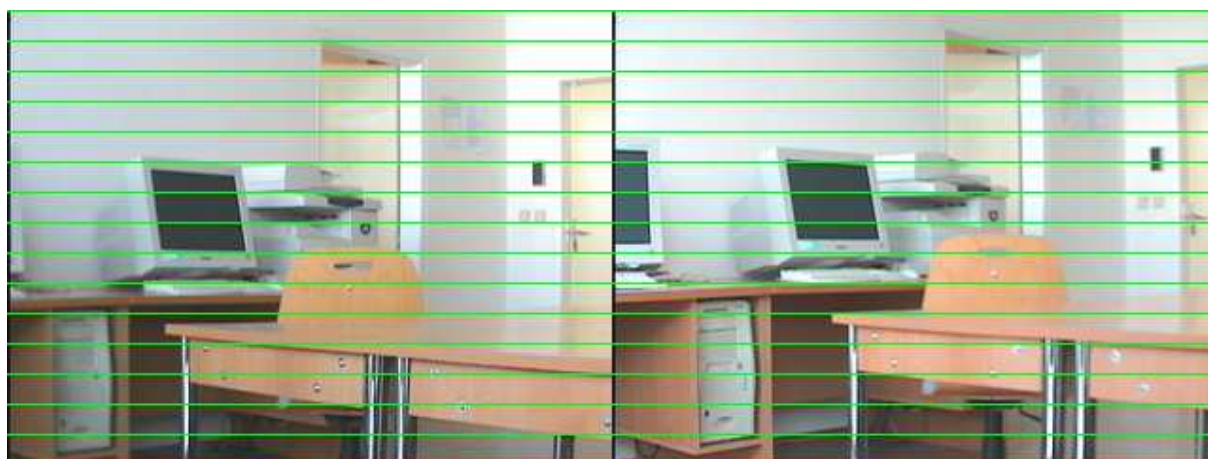
Příloha 12: Hledání korespondencí na základě kalibračních obrazců



Příloha 13: Nekalibrovaná rektifikace



Nalezení korespondencí metodou SURF

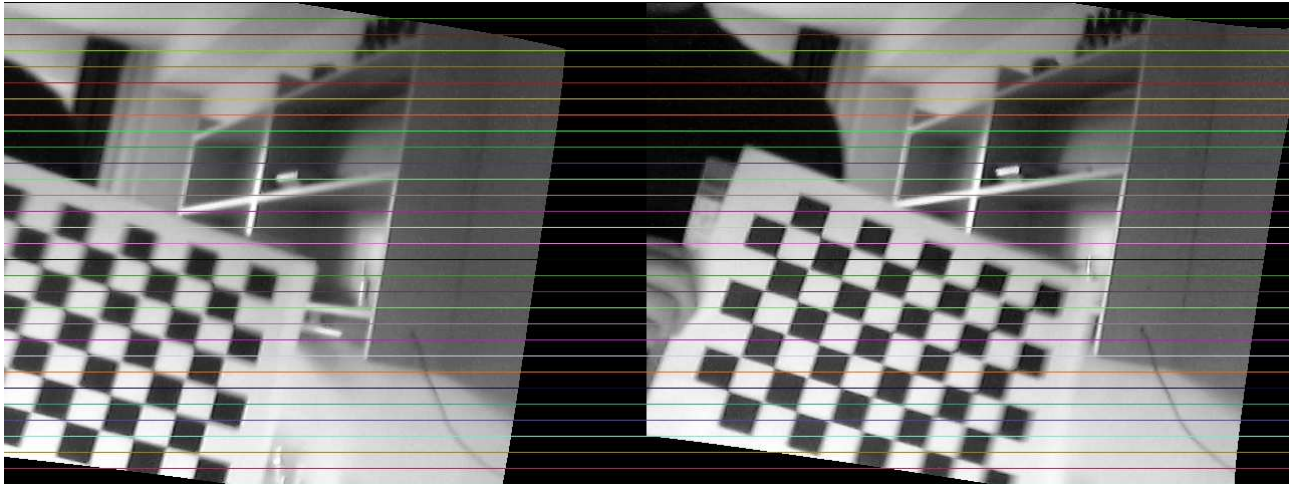


Původní snímky nejsou řádkově zarovnaný

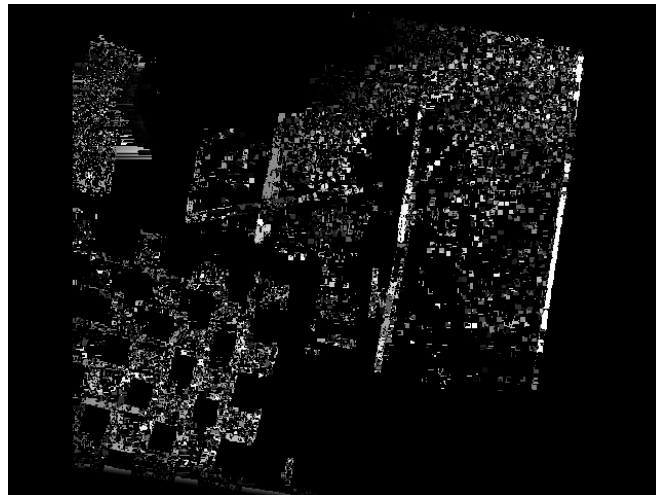


Rektifikované snímky jsou řádkově zarovnaný

Příloha 14: Kalibrovaná rektifikace



Rektifikované snímky z webové kamery



Disparitní mapa z rektifikovaných snímků

Příloha 15: Obsah CD

	ADRESÁŘ (/ - kořenový adresář)	POPIS
I.	/Bin/ /Bin/StereoCompute /Bin/PreviewCapture /Bin/Rectification /Bin/testsse2	Spustitelné soubory Výpočet stereoskopické báze Prohlížení a zpracování videa Test Rektifikace Test optimalizace vybraných metod
II.	/Projekty/ /Projekty/StereoCompute /Projekty/PreviewCapture /Projekty/Rectification /Projekty/testsse2	Zdrojové soubory projektů pro Visual studio2008
III.	/Video/ /Video/Vytvořené /Video/Kamery /Video/Webkamery /Video/Internet	Testovací videa Video vytvořené v 3D studiu Blender Záznam z kamer Záznam z webových kamer Stereoskopická videa z internetu
IV.	/Dokumenty/	Obrázky a grafy k diplomové práci
V.	/manual.pdf /pouziti.pdf /xhasma01.pdf /README.txt	Manuál k aplikacím Příklad použití aplikací Text diplomové práce Popis o aplikaci, prostředí, v kterém byla vytvořena, a jak ji spustit