

Praktikum 11

Stereo Vision

1. Disparity Map (Offline)

Program berikut ini memproses dua buah gambar stereo menjadi sebuah gambar yang menunjukkan pemetaan perbedaan kedalaman (*disparity map*) secara offline.

```
#include "cv.h"
#include "highgui.h"
#include "cvaux.h"
#include "stdio.h"

int main (int argc, char ** argv)
{
    IplImage* srcLeft = cvLoadImage("left.jpg",1);
    IplImage* srcRight = cvLoadImage("right.jpg",1);
    IplImage* leftImage = cvCreateImage(cvGetSize(srcLeft), IPL_DEPTH_8U, 1);
    IplImage* rightImage = cvCreateImage(cvGetSize(srcRight), IPL_DEPTH_8U, 1);
    IplImage* depthImage = cvCreateImage(cvGetSize(srcRight), IPL_DEPTH_8U, 1);

    cvCvtColor(srcLeft, leftImage, CV_BGR2GRAY);
    cvCvtColor(srcRight, rightImage, CV_BGR2GRAY);

    cvFindStereoCorrespondence( rightImage, leftImage, CV_DISPARITY_BIRCHFIELD,
    depthImage, 255, 15, 3, 6, 8, 15 );

    cvNamedWindow("left", 1);
    cvNamedWindow("right", 1);
    cvNamedWindow("depth", 1);

    cvShowImage("left", leftImage);
    cvShowImage("right", rightImage);
    cvShowImage("depth", depthImage);

    cvWaitKey(0);

    cvSaveImage("depth.jpg", depthImage);
    cvReleaseImage(&depthImage);
    cvReleaseImage(&rightImage);
    cvReleaseImage(&leftImage);
    cvReleaseImage(&srcLeft);
    cvReleaseImage(&srcRight);
    return 0;
}
```

Petunjuk praktikum:

- Siapkan *stereo camera*, kemudian ambil gambar dari masing masing kamera (kanan dan kiri), berikan nama file gambar tersebut dengan nama "right.jpg" dan "left.jpg", dan jalankan program di atas. Jelaskan fungsi berikut ini beserta dengan parameter yang ada di dalamnya.

- o cvFindStereoCorrespondence ()

- Lakukan modifikasi program diatas dengan mengganti fungsi cvFindStereoCorrespondence () dengan fungsi dibawah ini:

- o cvFindStereoCorrespondenceBM () // block matching
 - o cvFindStereoCorrespondenceGC () // graph-cut

2. Disparity Map (Online)

Program berikut ini memproses dua buah camera stereo untuk dilakukan pemetaan perbedaan kedalaman (*disparity map*) secara online.

```
#include "cv.h"
#include "cxcore.h"
#include "cvaux.h"
#include "highgui.h"

#define FRAME_WIDTH 320
#define FRAME_HEIGHT 240

int main (int argc, char ** argv)
{
    cvNamedWindow( "CAM1", 1 );
    cvNamedWindow( "CAM2", 1 );
    cvNamedWindow( "DEPTH", 1 ); //CV_WINDOW_AUTOSIZE

    CvCapture* capture1 = cvCaptureFromCAM(0); //right
    CvCapture* capture2 = cvCaptureFromCAM(1); //left
    CvCapture* capture3 = cvCaptureFromCAM(0);

    cvSetCaptureProperty(capture1, CV_CAP_PROP_FRAME_WIDTH, FRAME_WIDTH);
    cvSetCaptureProperty(capture1, CV_CAP_PROP_FRAME_HEIGHT, FRAME_HEIGHT);
    cvSetCaptureProperty(capture2, CV_CAP_PROP_FRAME_WIDTH, FRAME_WIDTH);
    cvSetCaptureProperty(capture2, CV_CAP_PROP_FRAME_HEIGHT, FRAME_HEIGHT);
    cvSetCaptureProperty(capture3, CV_CAP_PROP_FRAME_WIDTH, FRAME_WIDTH);
    cvSetCaptureProperty(capture3, CV_CAP_PROP_FRAME_HEIGHT, FRAME_HEIGHT);

    IplImage* frame1;
    IplImage* img1;
    IplImage* gray1;

    IplImage* frame2;
    IplImage* img2;
    IplImage* gray2;

    IplImage* frame3;
    IplImage* img3;
    IplImage* gray3;
    IplImage* depth;

    while(1)
    {
        frame1 = cvQueryFrame( capture1 );
        frame2 = cvQueryFrame( capture2 );
        frame3 = cvQueryFrame( capture3 );

        img1 = cvCloneImage( frame1 );
        gray1 = cvCreateImage( cvSize(img1->width, img1->height), IPL_DEPTH_8U, 1 );
        cvCvtColor( img1, gray1, CV_BGR2GRAY );

        img2 = cvCloneImage( frame2 );
        gray2 = cvCreateImage( cvSize(img2->width, img2->height), IPL_DEPTH_8U, 1 );
        cvCvtColor( img2, gray2, CV_BGR2GRAY );

        img3 = cvCloneImage( frame3 );
        gray3 = cvCreateImage( cvSize(img3->width, img3->height), IPL_DEPTH_8U, 1 );
        cvCvtColor( img3, gray3, CV_BGR2GRAY );

        depth = gray3;

        cvShowImage( "CAM1", gray1 );
        cvShowImage( "CAM2", gray2 );
        cvFindStereoCorrespondence( gray1, gray2, CV_DISPARITY_BIRCHFIELD,
depth, 250, 15, 3, 6, 8, 15 );
        cvShowImage( "DEPTH", depth );

        char c = cvWaitKey(33);
        if( c == 27 ) break;
    }

    cvWaitKey(0);
}
```

```
cvDestroyWindow( "CAM1" );
cvReleaseCapture( &capture1 );
cvReleaseImage( &img1 );
cvReleaseImage( &gray1 );
cvDestroyWindow( "CAM2" );
cvReleaseCapture( &capture2 );
cvReleaseImage( &img2 );
cvReleaseImage( &gray2 );
cvDestroyWindow( "DEPTH" );
cvReleaseImage( &depth );

return 0;
}
```

Petunjuk praktikum:

- Siapkan *stereo camera*, pastikan kedua kamera terkoneksi dengan computer dan berfungsi, kemudian jalankan program di atas secara online.
- Lakukan modifikasi program diatas dengan mengganti fungsi `cvFindStereoCorrespondence()` dengan fungsi dibawah ini:

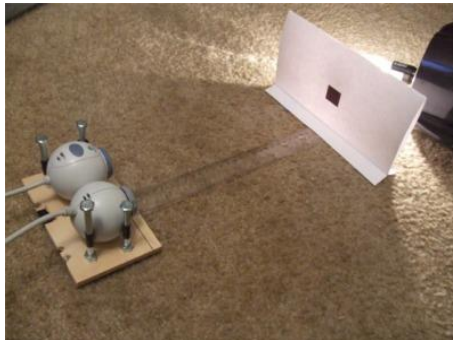
- `cvFindStereoCorrespondenceBM()` // block matching
- `cvFindStereoCorrespondenceGC()` // graph-cut

Tugas:

Buatlah sebuah program yang dapat mengukur jarak. Output dari program tersebut adalah jarak dalam satuan tertentu (cm). Dengan menggunakan algoritma sebagai berikut:

- Canny edge detection
- Corner detection
- Calculate distance

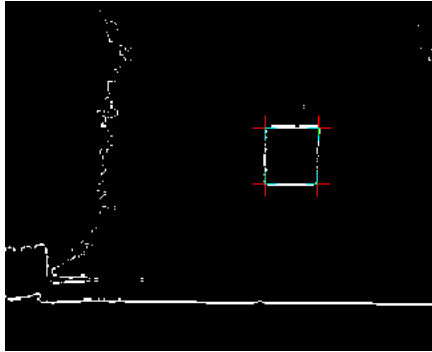
Berikut ini contoh aplikasi *distance measurement*.



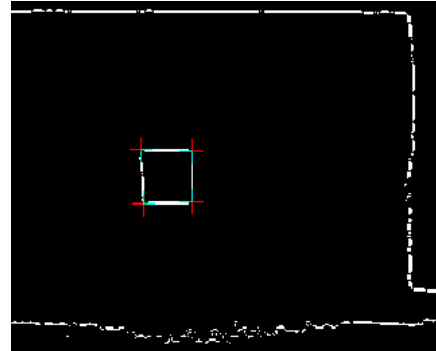
Gambar:

Sebuah objek diwakili kotak berwarna hitam dengan latar belakang putih.

Berikut ini adalah contoh hasil pengukuran jarak.



Top Left: 193,152 Top Right: 233,152
Bottom Left: 193,114 Bottom Right: 232,114



Top Left: 97,139 Top Right: 134,138
Bottom Left: 99,103 Bottom Right: 134,104

Distance: 27.865997 Shift: 94.000000

Source: <http://www.pages.drexel.edu/~nk752/distanceTut.html>