Nama = Muhammad Wafiq Kamaluddin

NRP = 2210181042

Kelas = 2 D4 TK B

Praktikum Deteksi Pojokan Dengan Metode Corner Harris

A. Tujuan.

1. mahasiswa mampu memahami Metode deteksi pojokan dengan metode corner harris.

2. mahasiswa mampu menerapkan metode corner harris pada bentuk program.

B. Dasar Teori.

Salah satu feature pada gambar adalah corner / sudut. Sudut dapat dideteksi dengan menggunakan metode Harris.  Metode ini menghitung selisih dari perpindahan pixel intensity dengan window disekelilingnya.

Anggap kita mempunyai citra grayscale,   perbedaan (selisih) antara origianal window dan window tetangganya E(u,v) adalah;



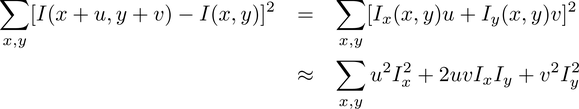
dengan u adalah displacement atau perpindahan pada arah x, v adalah perpindahan pada arah y. w(x,y) adalah window pada kordinat x dan y. I adalah intesitas piksel pada  kordinat (x,y). I (x+u,y+v) adalah pixel intensity pada kordinat (x+u, y+ v).

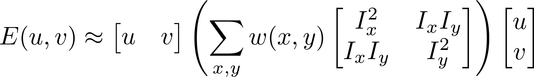
Rumus di atas cukup rumit untuk diselesaikan sehingga kita perlu menyederhanakan dengan menggunakan ekspansi Taylor. Khususnya pada bagian dalam kurung kurawal.



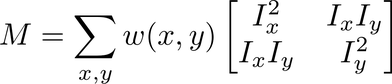
Dengan menggunakan ekspansi Taylor dua dimensi dan hanya mengambil sisi linear saja, kita dapat menyederhanakan persamaan di atas. Berikut adalah rumus ekspansi Taylor,

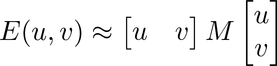
Dengan mengekspansi I(x+u, y+v) kita akan mendapatkan;

Dengan mensubstitusi persamaan di atas ke persamaan paling atas, serta menulis dalam notasi matriks kita akan mendapatkan



Selanjutnya variabel dalam kurng di atas akan dinotasikan sebagai M



Persamaannya menjadi 

Nilai R dihitung dengan menggunakan rumus



Nilai R tertentu dapat menentukan suatu daerah dianggap sudut atau bukan. Nilai  u, v dan k adalah parameter input.

C. Program.

#include "opencv2/core.hpp"

#include "opencv2/imgproc.hpp"

#include "opencv2/imgcodecs.hpp"

#include "opencv2/highgui.hpp"

#include <iostream>

using namespace cv;

using namespace std;

int main( int argc, char\*\* argv )

{

Mat src, src\_gray;

int blockSize = 10, apertureSize = 15, k = 0.02,thresh = 180;

src = imread("lap.jpg");

if ( src.empty() )

{

cout << "Could not open or find the image!\n" << endl;

return -1;

}

cvtColor( src, src\_gray, COLOR\_BGR2GRAY );

Mat dst = Mat::zeros( src.size(), CV\_32FC1 );

cornerHarris( src\_gray, dst, blockSize, apertureSize, k );

Mat dst\_norm, dst\_norm\_scaled;

normalize( dst, dst\_norm, 0, 255, NORM\_MINMAX, CV\_32FC1, Mat() );

convertScaleAbs( dst\_norm, dst\_norm\_scaled );

for( int i = 0; i < dst\_norm.rows ; i++ )

{

for( int j = 0; j < dst\_norm.cols; j++ )

{

if( (int) dst\_norm.at<float>(i,j) > thresh )

{

circle( dst\_norm\_scaled, Point(j,i), 5, 255, 2, 8, 0 );

}

}

}

imshow( "hasil deteksi", dst\_norm\_scaled );

imshow("gambar awal", src);

waitKey();

return 0;

}

D. Hasil

