공유 : 풀뿌리 시작 기업 의 프로젝트 소스 코드

FPGA 가속 : 컴퓨터 비전 (Computer Vision) AI 알고리즘

Learning through project!

프로젝트를 통해 학습

stereo vision 3D sensor

3Deepercept

내 시작 기업 기획서

percept the world deeper with stereo vision 3D sensor

Github:

https://github.com/brianwchh/grassrootsstartup-ComputerVsion-zynq

Youtube:

https://youtube.com/playlist?list=PL4mHdDqV3T2uioDIKB7c27LtLq5Z4AETl

Intel RealSense 와 동일한 제품

제품 시장

비디오 자습서

Grassroots Startup Project Sharing

Deep Learning Computer Vision 3D Sensor Algorithm Acceleration On FPGA Tutorial

4 – SGM cost function Dynamic Programming & Penalty Propagation

By GimHo.Ng/ 伍金和 (荷)

2023.3.29

G.E.M 鄧紫棋歌中的祕密 / the secret in GEM's music

【2507抬上帝入天坑】/【2507 carrying God back to heaven shit hole】: https://github.com/brianwchh/2507

Tutorial files: https://github.com/brianwchh/grassrootsstartup-ComputerVsion-zynq

소스 코드

```
matching_cost.v | LPDiLeftVolumn.v | rectify.v

/*
    author : WuChengHe

    funtion discription :
        Lr0(p,Di) = C(p,Di) + min(Lr0(p-1,Di),Lr0(p-1,Di-1)+P1,Lr0(p-1,Di+1)+P1,min(Di+/-k) + P2) - min(Lr0(p-1,Dk))
        compute L(p,Di) = Lr0(p,Di) + Lr1(p,Di)+ Lr2(p,Di)+ Lr3(p,Di)

*/
`timescale 1 ns / 1 ps
module LPDiLeftVolumn
#(
    parameter  IMAGE_WIDTH = 640 ,
    parameter  IMAGE_HEIGHT = 480 ,
    parameter  MAXDISPARITY = 64 ,
    parameter  PENALTY1 = 10 ,
    parameter  PENALTY2 = 60 ,
    parameter  CPDI_WIDTH = 6 ,
    parameter  LPDI_WIDTH = 8 ,
    parameter  INPUTDATAWID = 385,    // 6 * 64 + 1(SOF)
    parameter  OUTPUTDATAWID = 513    // 8 * 64 + 1
)
(
    input wire                          clk                 ,       // 150 Clock
    input wire                          en                  ,
    input wire                          pixelEN             ,       // clk / 8
    input wire                          sof_in              ,
    input wire                          eol_in              ,
    output wire                         sof_out             ,
    output wire                         eol_out             ,
    input  wire[INPUTDATAWID-1:0]       CPDi_1D             ,
    output wire[OUTPUTDATAWID-1:0]      LPDi                ,       // W*H*D volumn
    input wire                          rst_n                       // Asynchronous reset active low

);


reg [LPDI_WIDTH-1:0] LPDi_2DArray_reg_sclkrate [0:MAXDISPARITY-1] ;
integer i,j,kk,r,c;


/*  note that :
    no need buffer C(p,Di), need linebuffer Lr(p-1,Di) (r=1,2,3) , direction r=1,2,3 need 640 bytes of blockram
    direction r=0 , no need linebuffer
*/

/*

    +-------------+-------------+-------------+-------------
    | P(x-1,y-1)  | P(x,y-1)    | P(x+1,y-1   | .............
    +-------------+-------------+-------------+-------------
    | P(x-1,y)    | P(x,y)      |             | .............
    +-------------+-------------+-------------+-------------

*/
/* ***********************************************************************************************************
```