TI DSP, MCU, Xilinx Zynq FPGA 기반의 프로그래밍 전문가 과정

<영상 처리>

2018.06.28 – 82일차

강사 – **Innova Lee(이상훈)**
gcccompil3r@gmail.com

학생 – 안상재
sangjae2015@naver.com

<opencv 기반의 다양한 영상처리 프로그래밍>

1. buttondown.cpp

```
#include "opencv2/opencv.hpp"

using namespace cv;
using namespace std;

void onMouse(int event, int x, int y, int flags, void *param)
{
   Mat *pMat = (Mat *) param;
   Mat image = Mat(*pMat);

   switch(event)
   {
      case EVENT_LBUTTONDOWN:
         if(flags & EVENT_FLAG_SHIFTKEY)
            rectangle(image, Point(x-5, y-5), Point(x+5, y+5), Scalar(255,0,0));
         else
            circle(image, Point(x,y),5, Scalar(0,0,255), 5);
         break;
      case EVENT_RBUTTONDOWN:
         circle(image, Point(x,y),5, Scalar(255,0,0), 5);
         break;
      case EVENT_LBUTTONDBLCLK:
         image = Scalar(255,255,255);
         break;
   }

   imshow("dstImage", image);
}

int main(void)
{
   Mat dstImage(512,512, CV_8UC3, Scalar(255,255,255));

   imshow("dstImage", dstImage);
   setMouseCallback("dstImage", onMouse, (void *)&dstImage);

   waitKey();

   return 0;
}
```

2. clipline.cpp

```
#include "opencv2/opencv.hpp"

using namespace cv;
using namespace std;

int main(void)
{
   Mat dstImage(512, 512, CV_8UC3, Scalar(255,255,255));

   if(dstImage.empty())
      return -1;

   Rect imgRect(100, 100, 300, 300);
   rectangle(dstImage, imgRect.tl(), imgRect.br(), Scalar(255,0,0),2);

   Point pt1(120, 50);
   Point pt2(300, 300);
   line(dstImage, pt1, pt2, Scalar(0,255,0),2);

   clipLine(imgRect, pt1, pt2);
   cout << "pt1 = " << pt1 << endl;
   cout << "pt2 = " << pt2 << endl;

   circle(dstImage, pt1, 5, Scalar(0,0,255),2);
   circle(dstImage, pt2, 5, Scalar(0,0,255),2);
   line(dstImage, pt1, pt2, Scalar(255,0,0),2);

   imshow("dstImage", dstImage);
   waitKey();

   return 0;
}
```

3. ellipse.cpp

```
#include "opencv2/opencv.hpp"

using namespace cv;
using namespace std;

int main(void)
{
   Mat dstImage(512, 512, CV_8UC3, Scalar(255,255,255));

   Point center(250,200);
   Size size(200, 100);

   rectangle(dstImage, Point(center.x - size.width, center.y - size.height),
           Point(center.x + size.width, center.y + size.height), Scalar(255,0,0));

   line(dstImage, Point(center.x - size.width, center.y),
      Point(center.x + size.width, center.y), Scalar(0,255,0));
   line(dstImage, Point(center.x, center.y - size.height),
      Point(center.x, center.y + size.height), Scalar(0,255,0));

   ellipse(dstImage, center, size, 0, 0, 360, Scalar(0, 0, 255));  // 처음 0도는 3시 방향
   ellipse(dstImage, center, size, 90, 0, 360, Scalar(0, 0, 255), 2); //

   RotatedRect box(center, size, 90);
   ellipse(dstImage, box, Scalar(255,0,0), 2);

   vector<Point> pts;
   ellipse2Poly(center, size, 90, 0, 360, 45, pts);    // 45도 만큼 끊어서 pts 에 저장해라
   polylines(dstImage, pts, true, Scalar(0,255,0), 4);

   Point pt1, pt2;

   for(int i=0; i<pts.size(); i++)
   {
      pt1 = pts[i];

      if(i == pts.size() - 1)
         pt2 = pts[0];
      else
         pt2 = pts[i+1];

      line(dstImage, pt1, pt2, Scalar(0,0,255));
   }

   imshow("dstImage", dstImage);
   waitKey();

   return 0;
}
```

4. line.cpp

- 직선 및 사각형 그리기

```
#include "opencv2/opencv.hpp"

using namespace cv;
using namespace std;

int main(void)
{
   Mat dstImage(512, 512, CV_8UC3, Scalar(255,255,255));

   rectangle(dstImage, Point(100, 100), Point(400, 400), Scalar(0,0,255),2);
   line(dstImage, Point(400,100),Point(100,400), Scalar(0,255,0));
   rectangle(dstImage, Point(400/2,100/2), Point(100/2, 400/2), Scalar(255,0,0));
   imshow("dstImage", dstImage);
   waitKey();

   return 0;
}
```

5. iterate.cpp

```
#include <iostream>
#include <iomanip>
#include <vector>

using namespace std;

int main(void)
{
   vector<vector<int>> M1(2, vector<int>(3,0));

   M1[0][0] = 10;
   M1[0][1] = 20;
   M1[0][2] = 30;
   M1[1][0] = 40;
   M1[1][1] = 50;
   M1[1][2] = 60;
   cout << "M1.size() = " << M1.size() << endl;
   cout << "M1[0].size() = " << M1[0].size() << endl;

   vector<vector<int>>::iterator it1;
   vector<int>::iterator it2;
   cout << endl << "iterator : " << endl;


   for(it1 = M1.begin(); it1 != M1.end(); it1++)
   {
      for(it2 = (*it1).begin(); it2 != (*it1).end(); it2++)
      {
         cout << setw(4) << *it2 << " ";
      }
      cout << endl;
   }

   return 0;
}
```