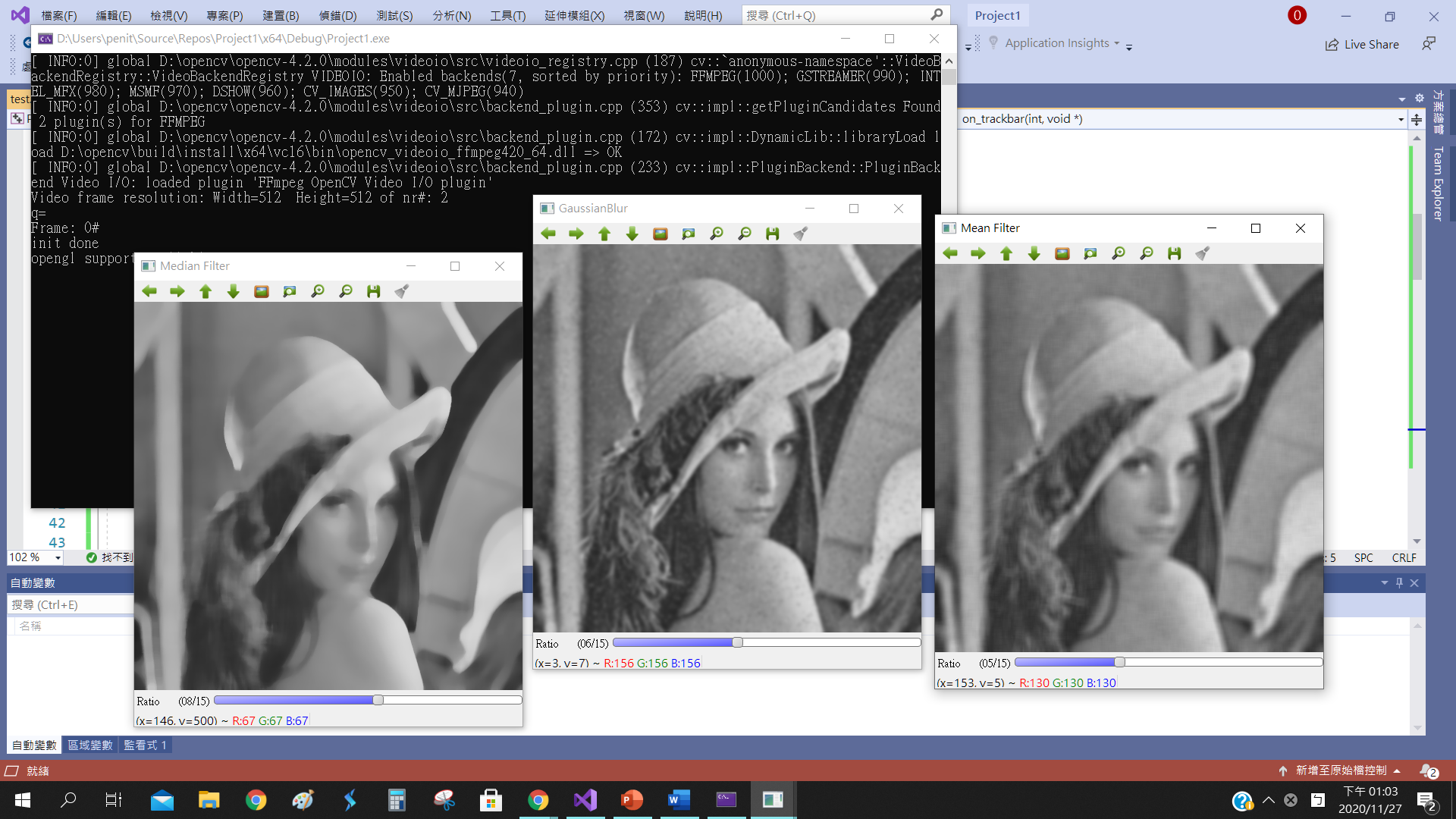
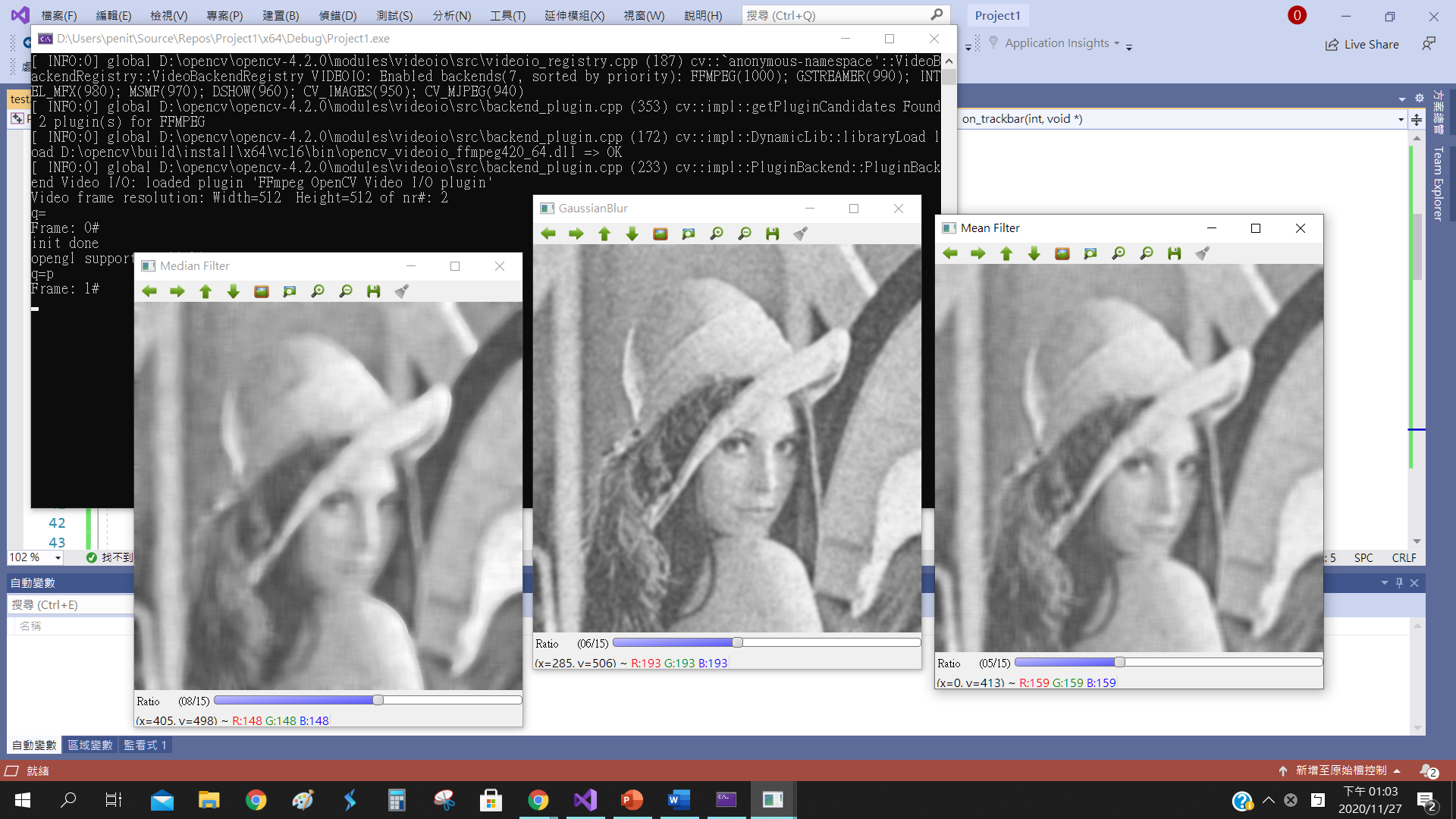
一開始

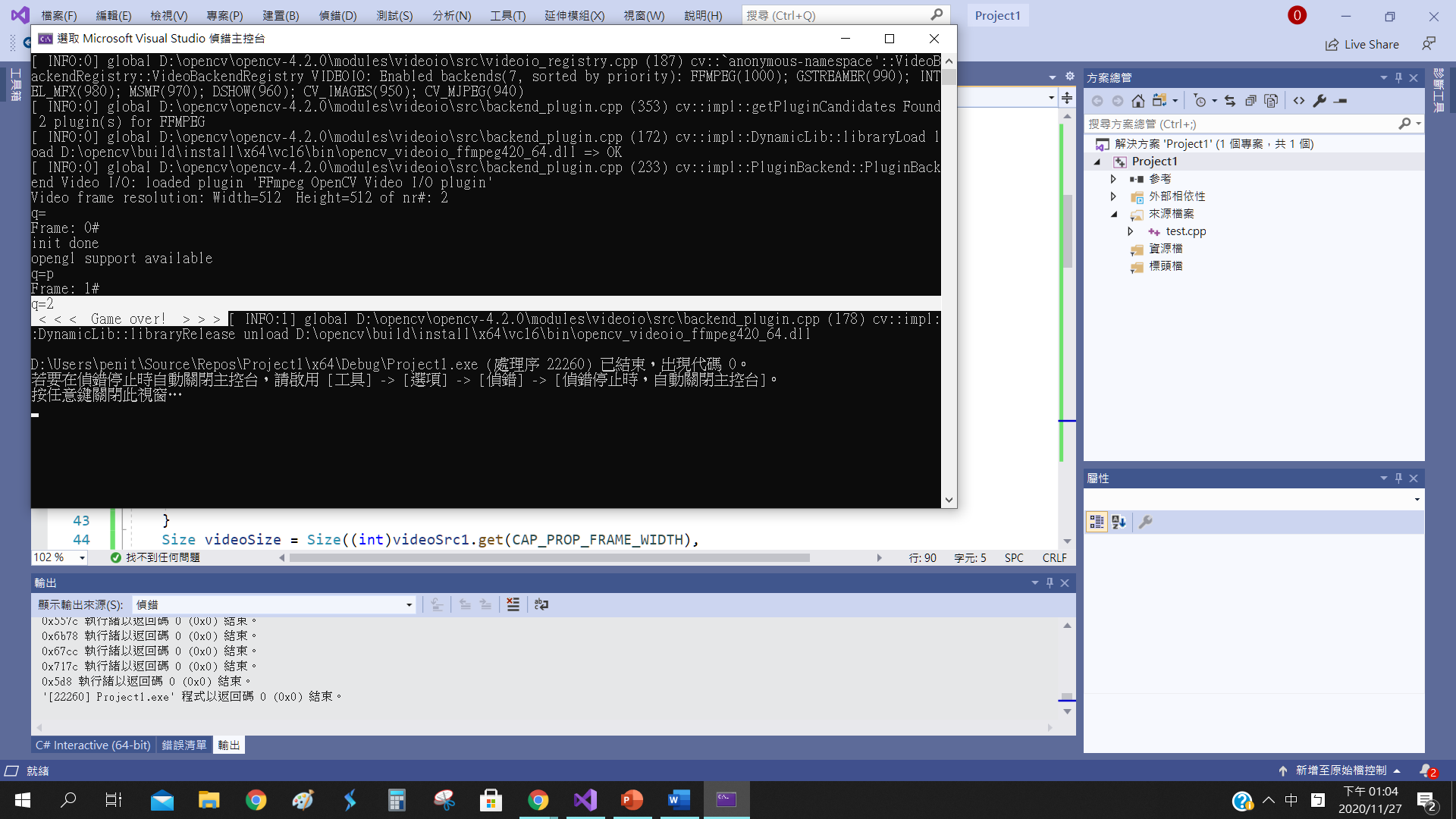


按任意鍵換下一張



再按任意鍵結束

或是一開始直接按 ’q’ 結束



|  |
| --- |
| /\*\*  \* file Smoothing.cpp  \* brief Sample code for simple filters  \* author OpenCV team  \*/  #include <iostream>  #include "opencv2/imgproc.hpp"  #include "opencv2/imgcodecs.hpp"  #include "opencv2/highgui.hpp"  #include <iostream> // for standard I/O  #include <string> // for strings  #include <opencv2/opencv.hpp>  using namespace std;  using namespace cv;  Mat src; Mat dst;  int sliderValue1, sliderValue2, sliderValue3;  bool applysave = false;  void on\_trackbar(int, void\*);  int main(int argc, char\* argv[])  {  sliderValue1 = 0;  sliderValue2 = 0;  sliderValue3 = 0;  int sliderMaxValue = 15;  string videofile;  int frameNum = -1; // Frame counter  int source;    VideoCapture videoSrc1;    //videofile = "images\\sequence\\Megamind\_%03d.jpg";  videofile = "lena\_noise%03d.jpg";  videoSrc1.open(videofile);      if (!videoSrc1.isOpened())  {  cout << "Could not open " << videofile << endl;  return -1;  }  Size videoSize = Size((int)videoSrc1.get(CAP\_PROP\_FRAME\_WIDTH),  (int)videoSrc1.get(CAP\_PROP\_FRAME\_HEIGHT));      cout << "Video frame resolution: Width=" << videoSize.width << " Height=" << videoSize.height  << " of nr#: " << videoSrc1.get(CAP\_PROP\_FRAME\_COUNT) << endl;  Mat frame;  for (;;) //Show the image captured in the window and repeat  {  char c = (char)waitKey();  if (c == 'q') break;  cout << "q=";  cout << c << endl;  videoSrc1.read(frame); // videoSrc1 >> frame;  if (frame.empty())  {  cout << " < < < Game over! > > > ";  break;  }  ++frameNum;  cout << "Frame: " << frameNum << "# " << endl;  src = frame;  dst = src.clone();  imshow("Mean Filter", src);  imshow("GaussianBlur", src);  imshow("Median Filter", src);  createTrackbar("Ratio", "Mean Filter", &sliderValue1, sliderMaxValue, on\_trackbar);  on\_trackbar(sliderValue1, 0);  createTrackbar("Ratio", "GaussianBlur", &sliderValue2, sliderMaxValue, on\_trackbar);  on\_trackbar(sliderValue2, 0);  createTrackbar("Ratio", "Median Filter", &sliderValue3, sliderMaxValue, on\_trackbar);  on\_trackbar(sliderValue3, 0);  //imshow("Video", frame);  }  return 0;  }  void on\_trackbar(int, void\*) {  int kernal1 = sliderValue1 \* 2 + 1;  int kernal2 = sliderValue2 \* 2 + 1;  int kernal3 = sliderValue3 \* 2 + 1;  blur(src, dst, Size(kernal1, kernal1), Point(-1, -1));  imshow("Mean Filter", dst);  GaussianBlur(src, dst, Size(kernal2, kernal2), 0, 0);  imshow("GaussianBlur", dst);  medianBlur(src, dst, kernal3);  imshow("Median Filter", dst);  } |