#include <chrono>

#include <opencv2/opencv.hpp>

#include <iostream>

#include <time.h>

using namespace cv;

using namespace std;

using namespace std::chrono;

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

{

VideoCapture cap(0);

if (cap.isOpened() == false)

{

cout << "Cannot open the video camera" << endl;

cin.get();

return -1;

}

double dWidth = cap.get(CAP\_PROP\_FRAME\_WIDTH);

double dHeight = cap.get(CAP\_PROP\_FRAME\_HEIGHT);

cout << "Resolution of the video : " << dWidth << " x " << dHeight << endl;

string window\_name = "My Camera Feed";

namedWindow(window\_name);

while (true)

{

auto start = steady\_clock::now();

Mat frame;

bool bSuccess = cap.read(frame);

if (bSuccess == false)

{

cout << "Video camera is disconnected" << endl;

cin.get();

break;

}

auto end1 = steady\_clock::now();

auto elapsed1 = duration\_cast<milliseconds>(end1 - start);

cout << "time of reading --- " << elapsed1.count() <<" ms"<< endl;

Mat new\_frame;

//frame.convertTo(new\_frame, -1, 0.5, +50);

for (int i = 0; i < frame.rows; i++)

for (int j = 0; j < frame.cols; j++)

{

frame.at<Vec3b>(i, j)[2] -= 20;

frame.at<Vec3b>(i, j)[1] -=20;

//frame.at<Vec3b>(i, j)[3] -=20;

}

frame.convertTo(new\_frame, -1, 2, +45);

auto end2 = steady\_clock::now();

auto elapsed2 = duration\_cast<milliseconds>(end2 - end1);

cout << "time of convertion --- " << elapsed2.count() << " ms" << endl;

imshow(window\_name, new\_frame);

auto end3 = steady\_clock::now();

auto elapsed3 = duration\_cast<milliseconds>(end3 - end2);

cout <<"time of output --- "<< elapsed3.count() << " ms" << endl;

if (waitKey(50) == 27)

{

cout << "Esc key is pressed by user. Stoppig the video" << endl;

break;

}

auto end4 = steady\_clock::now();

auto elapsed4 = duration\_cast<milliseconds>(end4 - start);

cout << "frame time --- " << 1000/elapsed4.count() << endl << endl;

}

return 0;

}