#include <opencv2/opencv.hpp>

#include <opencv2/highgui/highgui.hpp>

#include <opencv2/nonfree/nonfree.hpp>

#include <opencv2/features2d/features2d.hpp>

using namespace *cv*;

using namespace *std*;

int *main*()

{

*system*("color 5F");

*Mat* trainImage=*imread*("1.jpg"),trainImage\_gray;

*imshow*("trainImage",trainImage);

*cvtColor*(trainImage,trainImage\_gray,*CV\_BGR2GRAY*);

*vector*<*KeyPoint*> train\_keyPoint;

*Mat* trainDescription;

*SiftFeatureDetector* featureDetector;

featureDetector.*detect*(trainImage\_gray,train\_keyPoint);

*SiftDescriptorExtractor* featureExtractor;

featureExtractor.*compute*(trainImage\_gray,train\_keyPoint,trainDescription);

*BFMatcher* matcher;

*vector*<*Mat*> train\_desc\_collection(1,trainDescription);

matcher.*add*(train\_desc\_collection);

matcher.*train*();

*VideoCapture* cap(0);

unsigned int frameCount=0;

while (char(*waitKey*(1))!='q')

{

double time0=*getTickCount*();

*Mat* captureImage,captureImage\_gray;

cap>>captureImage;

if(captureImage.*empty*())

continue;

*cvtColor*(captureImage,captureImage\_gray,*CV\_BGR2GRAY*);

*vector*<*KeyPoint*> test\_keyPoint;

*Mat* testDescriptor;

featureDetector.*detect*(captureImage\_gray,test\_keyPoint);

featureExtractor.*compute*(captureImage\_gray,test\_keyPoint,testDescriptor);

*vector*<*vector*<*DMatch*>> matches;

matcher.*knnMatch*(testDescriptor,matches,2);

*vector*<*DMatch*> goodMatches;

for (unsigned int i=0;i<matches.*size*();i++)

{

if(matches[i][0].*distance*<0.6\*matches[i][1].*distance*)

goodMatches.*push\_back*(matches[i][0]);

}

*Mat* dstImage;

*drawMatches*(captureImage,test\_keyPoint,trainImage,train\_keyPoint,goodMatches,dstImage);

*imshow*("dstImage",dstImage);

*cout*<<">>当Ì¡À前¡ã频¦Ì率¨º为a：êo"<<*getTickFrequency*()/(*getTickCount*()-time0)<<*endl*;

}

return 0;

}