



faq

tags

people

badges

ALL

UNANSWERED

Hi, there! Please sign in help

ASK YOUR QUESTION

Understanding the blobtrack_sample.cpp modules

0

tracking

opencv

modules

blob

Hi i am trying to understand how the tracking modules of the blobtrack_sample.cpp example are working. For that reason and after quite a lot of research i managed to create an example with my own foreground detector. However, although i am able to detect and follow the blobs on the screen i cannot understand how to draw the trajectories by using the BlobTrackGen module does someone has any idea in order to point me to a direction on how to do it. Here is my code:

asked Jan 17 '13
theodore
151 • 3 • 5

updated Jan 17 '13

8+1

```
#include <vector>
#include <opencv/cv.h>
#include <opencv/cvaux.h>
#include <opencv/highgui.h>
#include <iostream>
#include <list>

#include <opencv2/video/background_segm.hpp>
#include <opencv2/legacy/blobtrack.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc_c.h>

#include <opencv2/core/core.hpp>
#include <opencv2/video/tracking.hpp>
```

```
using namespace std;
using namespace cv;
```

```
class SimpleDetector : public CvF6Detector
{
    IplImage * mask;
public:
```

```
    SimpleDetector()
    {
        mask = 0;
        SetTypeName("SD");
    }
```

```
virtual IplImage* GetMask()
{
    return mask;
}
```

```
virtual void Process(IplImage* img)
{
    Mat frame(img);
    Mat thresh_frame;
    vector<Mat> channels;
    vector<Vec4i> hierarchy;
    vector<vector<Point>> contours;

    split(frame, channels);
    add(channels[0], channels[1], channels[1]);
    subtract(channels[2], channels[1], channels[2]);
    threshold(channels[2], thresh_frame, 50, 255, CV_THRESH_BINARY);
    medianBlur(thresh_frame, thresh_frame, 5);

    findContours(thresh_frame, contours, hierarchy, CV_RETR_EXTERNAL, CV_CHAIN_APPROX_SIMPLE, I

    Mat drawing = Mat::zeros(thresh_frame.size(), CV_8UC1);
    for(size_t i = 0; i < contours.size(); i++)
    {
        if(contourArea(contours[i]) > 500)
            drawContours(drawing, contours, i, Scalar::all(255), CV_FILLED, 8, vector<Vec4i>(), 0,
```

Question tools

Follow

☐ Here (once you log in) you will be able to sign up for the periodic email updates about this question.

[subscribe to rss feed](#)

Stats

Asked: Jan 17 '13

Seen: 361 times

Last updated: Jan 17 '13

Related questions

[Removing outliers from goodFeaturesToTrack using the x84 method](#)

[OpenCV Image blob detection and Gesture support on Android](#)

[Cannot create an OpenGL window](#)

[Problems installing opencv on mac with python](#)

[parse error](#)

[Problem with c++ and cv::resize](#)

[OpenCV and ueye-camera](#)

[opencv 2.4.3 iOS background_segm.hpp 'list' file not found](#)

[Human height estimation using a calibrated camera](#)

[cvCompareHist error](#)

```

    }
    thresh_frame = drawing;

    findContours(thresh_frame, contours, hierarchy, CV_RETR_EXTERNAL, CV_CHAIN_APPROX_SIMPLE, 1);

    drawing = Mat::zeros(thresh_frame.size(), CV_8UC1);
    for(size_t i = 0; i < contours.size(); i++)
    {
        if(contourArea(contours[i]) > 500)
            drawContours(drawing, contours, i, Scalar::all(255), CV_FILLED, 8, vector<Vec4i>(), 0);
    }
    thresh_frame = drawing;

    IplImage tmp = thresh_frame;

    if (!mask)
        mask = cvCreateImage(cvGetSize(&tmp), tmp.depth, 1);

    cvCopy(&tmp, mask);
}

/* Release foreground detector: */
virtual void Release()
{
    if (mask)
        cvReleaseImage(&mask);
}
};

int main(int argc, char** argv)
{
    CvCapture* cam = NULL;
    cam = cvCreateCameraCapture(0);

    cvNamedWindow("Original", CV_WINDOW_AUTOSIZE);
    cvNamedWindow("Mask", CV_WINDOW_AUTOSIZE);
    cvNamedWindow("Mask_v1", CV_WINDOW_AUTOSIZE);
    cvNamedWindow("Final", CV_WINDOW_AUTOSIZE);

    //+++++

    CvBlobTrackerAutoParam1 params;
    CvBlobTrackerAuto* tracker;

    SimpleDetector sd;

    params.pFG = &sd;
    params.FGTrainFrames = 0;
    params.pBD = cvCreateBlobDetectorSimple();
    params.pBT = cvCreateBlobTrackerMSPF();
    params.pBTA = cvCreateModuleBlobTrackAnalysisHistPVS();
    params.pBTGen = cvCreateModuleBlobTrackGen1();
    // params.pBTGen->SetFileName("trajectories.txt");
    params.pBTPP = cvCreateModuleBlobTrackPostProcKalman();

    tracker = cvCreateBlobTrackerAuto1(&params);

    //+++++

    IplImage * _img = cvQueryFrame(cam);
    while (true)
    {
        _img = cvQueryFrame(cam);

        CvSize sz = cvSize(_img->width, _img->height);
        IplImage* _img2 = cvCreateImage(sz, 8, 3);
        IplImage * _maskImg = cvCreateImage(sz, 8, 1);
        cvResize(_img, _img2);
        sd.Process(_img2);
        IplImage* _maskImgTemp = sd.GetMask();
        cvResize(_maskImgTemp, _maskImg);
        IplImage * _fImg = cvCreateImage(sz, 8, 3);
        cvZero(_fImg);

        tracker->Process(_img2, /*NULL*/_maskImg);
    }
}

```

```

// cout << tracker->GetBlobNum() << endl;

if (tracker->GetBlobNum() > 0)
{
    char str[1024];
    CvFont font;
    int line_type = CV_AA;    // Change it to 8 to see non-antialiased graphics.

    cvInitFont( &font, CV_FONT_HERSHEY_PLAIN, 0.7, 0.7, 0, 1, line_type );
    for (int i = tracker->GetBlobNum(); i > 0; i--)
    {
        CvSize TextSize;
        CvBlob* pB = tracker->GetBlob(i-1);
        CvPoint p = cvPoint(cvRound(pB->x*256), cvRound(pB->y*256));
        CvSize s = cvSize(MAX(1, cvRound(CV_BLOB_RX(pB)*256)), MAX(1, cvRound(CV_BLOB_RY(pB)*256)));
        int c = cvRound(255*tracker->GetState(CV_BLOB_ID(pB)));

        cvEllipse( _img2,
                    p,
                    s,
                    0, 0, 360,
                    CV_RGB(c,255-c,0), cvRound(1+(3*0)/255), CV_AA, 8 );

        p.x >>= 8;
        p.y >>= 8;
        s.width >>= 8;
        s.height >>= 8;
        sprintf(str, "%03d", CV_BLOB_ID(pB));
        cvGetTextSize( str, &font, &TextSize, NULL );
        p.y -= s.height;
        cvPutText( _img2, str, p, &font, CV_RGB(0,255,255));
    }
}

cvShowImage("Original", _img2);
cvShowImage("Mask", _maskImgTemp);
cvShowImage("Mask_v1", _maskImg);
cvShowImage("Final", _fImg);

char key = cvWaitKey(3);
if (key == 'q')
{
    sd.Release();
    cvReleaseImage(&_img2);
    cvReleaseImage(&_maskImgTemp);
    cvReleaseImage(&_maskImg);
    cvReleaseImage(&_fImg);
    break;
}
}

cvDestroyAllWindows();
if (cam)cvReleaseCapture(&cam);
if (params.pBT)cvReleaseBlobTracker(&params.pBT);
if (params.pBD)cvReleaseBlobDetector(&params.pBD);
if (params.pBTGen)cvReleaseBlobTrackGen(&params.pBTGen);
if (params.pBTA)cvReleaseBlobTrackAnalysis(&params.pBTA);
if (params.pFG)cvReleaseFGDetector(&params.pFG);
if (tracker)cvReleaseBlobTrackerAuto(&tracker);

return 0;
}

```

p.s. the subtractor that i have made is just extracting the red objects, so if you want to use the above code you need a red object in front of your camera



Comments

no one has any idea, or worked before with the blobtrack_sample :-(
[theodore](#) (Jan 30 '13) [edit](#)

[Login/Signup to Answer](#)

[about](#) | [faq](#) | [help](#) | [privacy policy](#) | [give feedback](#)
Powered by Askbot version 0.7.43

Copyright Itseez, 2013. Content on this site is licensed under
a Creative Commons Attribution Share Alike 3.0 license.