【OpenCV/aruco】创建board Demo

o0o_-_ 于 2019-04-10 17:41:32 发布

- **说在前面**

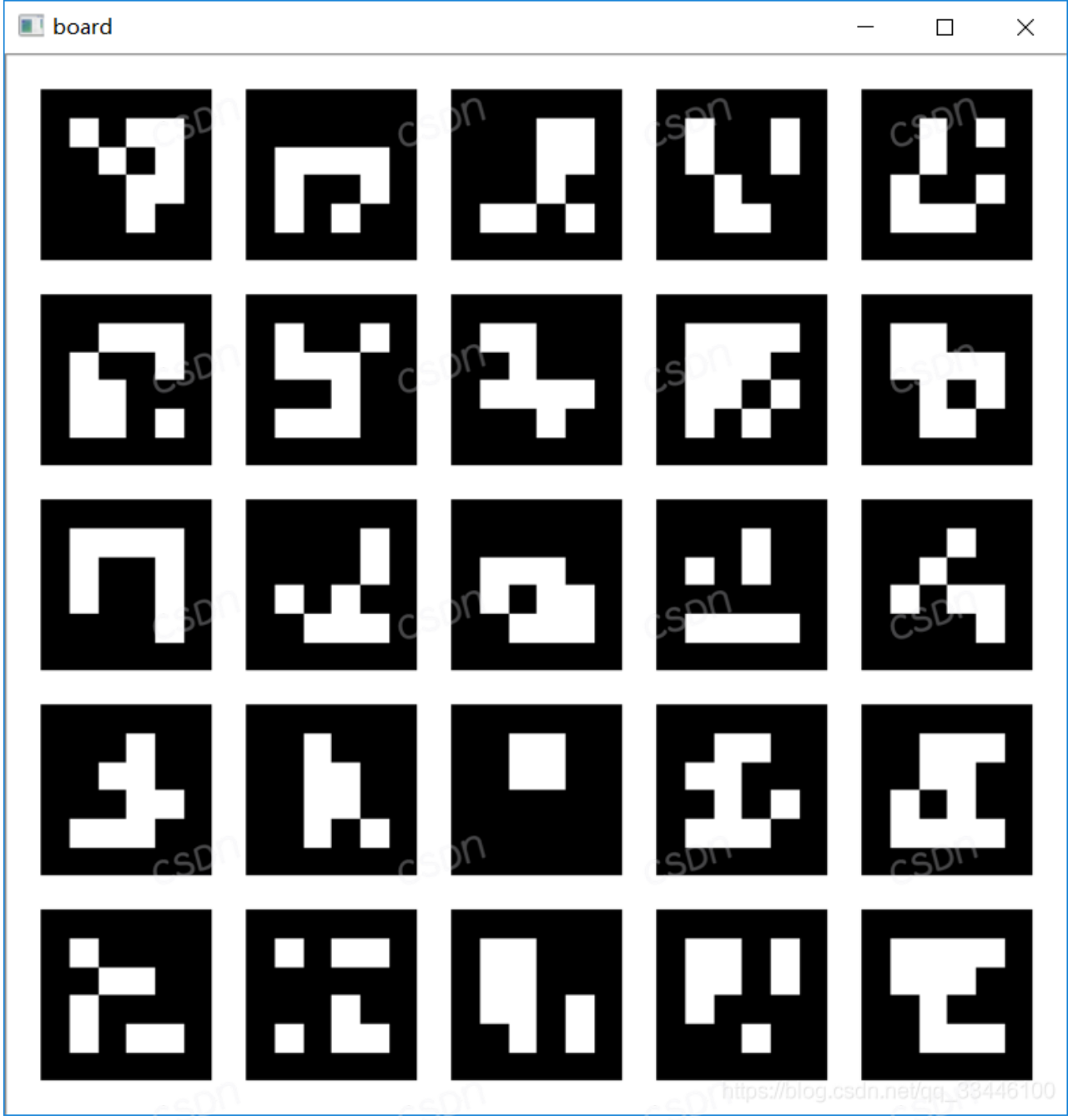
> 操作系统：win10
>
> vs 版本：2017
>
> opencv版本：4.0.1
>
> opencv-contrb版本：4.0.1
>
> 接上篇：【OpenCV/aruco】检测marker Demo
>
> 下一篇：【OpenCV/aruco】校准相机(Camera Calibration) Demo

- **Board**

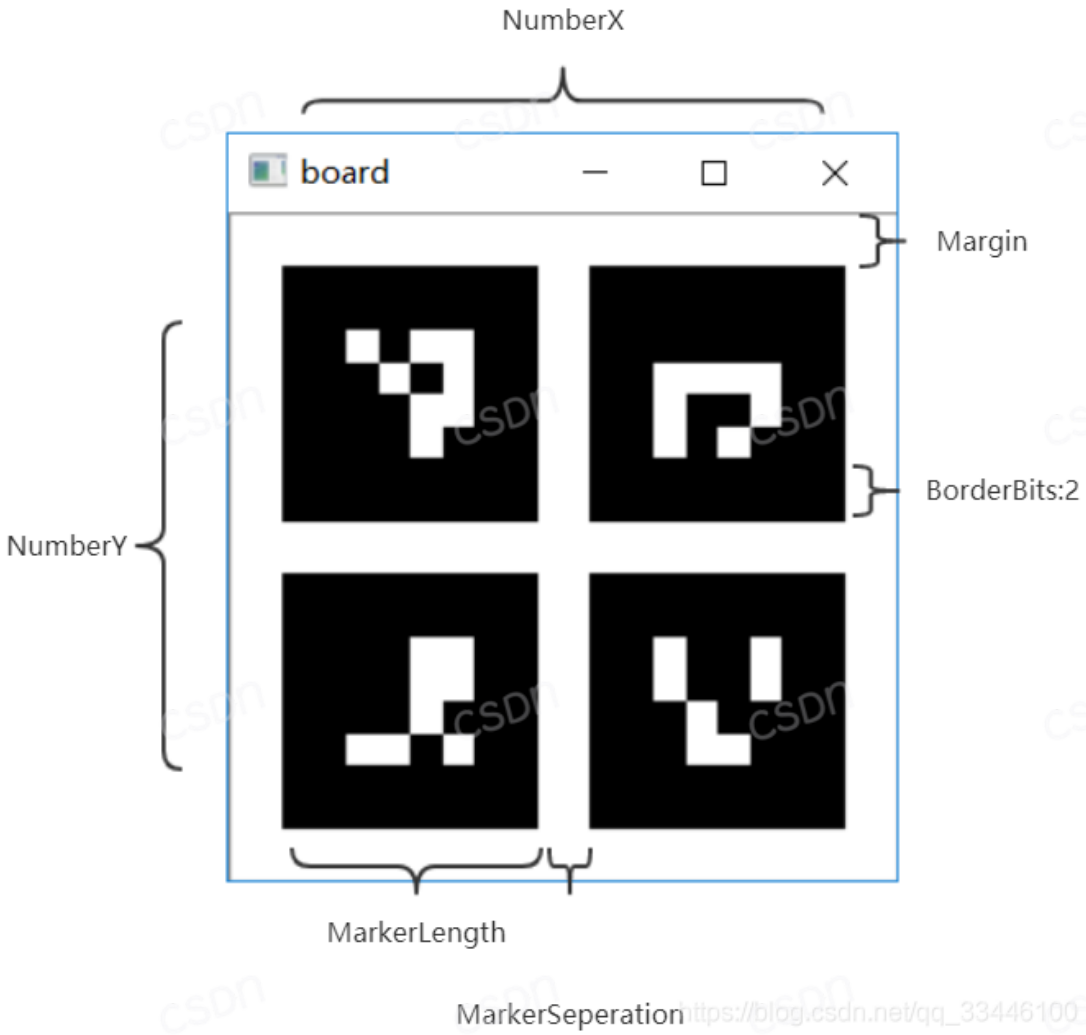
  board是一个aruco marker的棋盘，上面布置了许多marker。

  board可用于校准（Calibrate）摄像机

- **运行截图**



- 说明



- **函数定义**

```
static Ptr<GridBoard> cv::aruco::GridBoard::create  (
int     markersX,          // X方向marker数量，即上图NumberX
int     markersY,          // Y方向marker数量，即上图NumberY
float   markerLength,      // marker长度，即上图MarkerLength
float   markerSeparation,  // marker之间的间隔，即上图MarkerSeperation
const Ptr< Dictionary > &   dictionary, // 字典
int     firstMarker = 0    //grid board上第一个marker的ID
                           //通过改变该参数可以生成不同的board
)
```

```
void cv::aruco::GridBoard::draw (
Size    outSize,        // 输出图像大小
OutputArray     img,    // 输出图像
int     marginSize = 0, // 即上图Margin，即最外面的marker与图像边界之间的距离
int     borderBits = 1  // 即上图BoderBits,代表每个marker边框大小
)
```

- 代码

```
#include <opencv2\highgui.hpp>
#include <opencv2\aruco.hpp>
#include <opencv2\aruco\dictionary.hpp>
#include <opencv2\aruco\charuco.hpp>
#include <opencv2\core.hpp>
#include <opencv2\imgproc\imgproc.hpp>
#include <opencv2\opencv.hpp>
#include <vector>
#include <iostream>

using namespace std;
using namespace cv;

int main()
{
    int markersX = 5;//X轴上标记的数量
    int markersY = 5;//Y轴上标记的数量    本例生成5x5的棋盘
    int markerLength = 100;//标记的长度，单位是像素
    int markerSeparation = 20;//每个标记之间的间隔，单位像素
    int dictionaryId = cv::aruco::DICT_4X4_50;//生成标记的字典ID
    int margins = markerSeparation;//标记与边界之间的间隔

    int borderBits = 1;//标记的边界所占的bit位数
    bool showImage = true;

    Size imageSize;
    imageSize.width = markersX * (markerLength + markerSeparation) - markerSeparation + 2 * margins;
    imageSize.height =
```

```
        markersY * (markerLength + markerSeparation) - markerSeparation + 2 * margins;
    Ptr<aruco::Dictionary> dictionary =
        aruco::getPredefinedDictionary(aruco::PREDEFINED_DICTIONARY_NAME(dictionaryId));

    Ptr<aruco::GridBoard> board = aruco::GridBoard::create(markersX, markersY, float(markerLength),
        float(markerSeparation), dictionary);

    // show created board
    Mat boardImage;
    board->draw(imageSize, boardImage, margins, borderBits);

    if (showImage) {
        imshow("board", boardImage);
        waitKey(0);
    }

    return 0;
}
```