

机器视觉实验

实验背景知识

代码分析

运行结果截图

结果分析

机器视觉实验

实验背景知识

球栅阵列（Ball Grid Array，简称BGA）封装技术为应用在集成电路上的一种表面黏着技术，此技术常用来永久固定如微处理器之类的装置。BGA封装能提供比其他如双列直插封装（Dual in-line package）或四侧引脚扁平封装（Quad Flat Package）所容纳更多的接脚，整个装置的底部表面可全作为接脚使用，而不是只有周围可使用，比起周围限定的封装类型还能具有更短的平均导线长度，以具备更佳的高速效能。

实验应用机器视觉技术,采集球栅阵列(BGA)芯片图像,对图像进行预处理后,运用点分析方法对图像中各个焊球区域进行标记,通过计算获得焊球数目以及各个焊球位置、大小等特征信息,建立判断标准并依次序对这些信息进行比对,从而判断芯片合格与否,以及出现问题的种类与数量。实验结果证明,此方法可以正确识别缺陷类型,在检测速度以及可靠性方面有很好的保证。

代码分析

```
*
* 这个程序演示了球栅阵列(BGA)的检查。首先，检查球的大小和形状。然后，检查BGA是否缺失，多余，及
* 位置错误。
*
* 1. 初始化程序
read_image (Image, 'bga/bga1_01')
dev_close_window ()
dev_open_window_fit_image (Image, 0, 0, 640, 480, windowHandle)
set_display_font (windowHandle, 16, 'mono', 'true', 'false')
dev_update_off ()
visualizeIntermediate := false
*
* 2. 设置分割参数和球的检查。
BrighterThan := 70
MinArea := 180
MaxArea := 230
MaxAnisometry := 1.2
MaxDistance := 0.003 * 25.4
*
* 3. 设置创建BGA模型的参数。
BgaLayout := [1,1,1,0,0,0,0,0,1,1,1,1]
BallDistRowRef := 0.05 * 25.4
BallDistColRef := 0.05 * 25.4
*
* 4. 分割球并检查其面积和形状(像素和亚像素精确)
for j := 1 to 6 by 1
    read_image (Image, 'bga/bga1_' + j$'02d')
```

```

segment_balls (Image, Balls, BrighterThan, MinArea, MaxArea) 、
check_balls (Balls, WrongAreaBalls, WrongAnisometryBalls, MinArea, MaxArea,
MaxAnisometry)、
display_checked_balls (Image, Balls, WrongAreaBalls, WrongAnisometryBalls)

dev_disp_text ('Press Run (F5) to continue', 'window', 'bottom', 'right',
'black', [], [])
stop ()
segment_balls_sub_pix (Image, BallsSubPix, BrighterThan, MinArea, MaxArea)
check_balls_sub_pix (BallsSubPix, WrongAreaBallsSubPix,
WrongAnisometryBallsSubPix, MinArea, MaxArea, MaxAnisometry)
display_checked_balls (Image, BallsSubPix, WrongAreaBallsSubPix,
WrongAnisometryBallsSubPix)
dev_disp_text ('Press Run (F5) to continue', 'window', 'bottom', 'right',
'black', [], [])
stop ()
endfor

```

* 5. 将球的实际位置与球的参考位置进行比较。

*

* 5.1 首先，创建并显示一个为球提供参考位置的BGA模型。

*

```

gen_bga_model (BgaLayout, BallDistRowRef, BallDistColRef, BallMatrixRef,
BallsRowsRef, BallsColsRef, BallsPerRow, BallsPerCol)
display_bga_matrix (BallsRowsRef, BallsColsRef, BallDistRowRef, BallDistColRef,
20, windowHandle)
dev_disp_text ('Press Run (F5) to continue', 'window', 'bottom', 'right',
'black', [], [])
stop ()

```

*

* 5.2 然后提取球的位置，并与参考位置进行比较。

for J := 1 to 6 by 1

```

    read_image (Image, 'bga/bga1_' + J$'02d')
    dev_display (Image)

```

*

* 5.3 分割球，检查他们的面积和形状。

```

segment_balls_sub_pix (Image, BallsSubPix, BrighterThan, MinArea, MaxArea)
check_balls_sub_pix (BallsSubPix, WrongAreaBallsSubPix,
WrongAnisometryBallsSubPix, MinArea, MaxArea, MaxAnisometry)

```

*

* 5.4 为当前检查的BGA创建BGA矩阵。

```

gen_bga_matrix_from_image (Image, BallsSubPix, BallsPerRow, BallsPerCol,
windowHandle, VisualizeIntermediate, BallsRows, BallsCols, BallDistCol,
BallDistRow, NumBalls, BallMatrix)

```

*

* 5.5 将球的图像坐标转换为BGA模型的参考坐标系。

```

transform_ball_coords_image_to_world (BallsSubPix, BallsPerRow, BallsPerCol,
BallMatrixRef, BallsRowsRef, BallsColsRef, BallMatrix, BallsRows, BallsCols,
windowHandle, VisualizeIntermediate, BallsMatchedRowsWorld,
BallsMatchedColsWorld, BallsMatchedRowsRef, BallsMatchedColsRef, IndexImage,
HomMat2DInvert)

```

*

* 5.6 确定提取的球和参考球之间的距离。

```

distance_pp (BallsMatchedRowsRef, BallsMatchedColsRef, BallsMatchedRowsWorld,
BallsMatchedColsWorld, Distances)

```

*

* 6. 将错误放置、丢失、多余的球以及错误形状的球的检查结果可视化。

```

dev_display (Image)

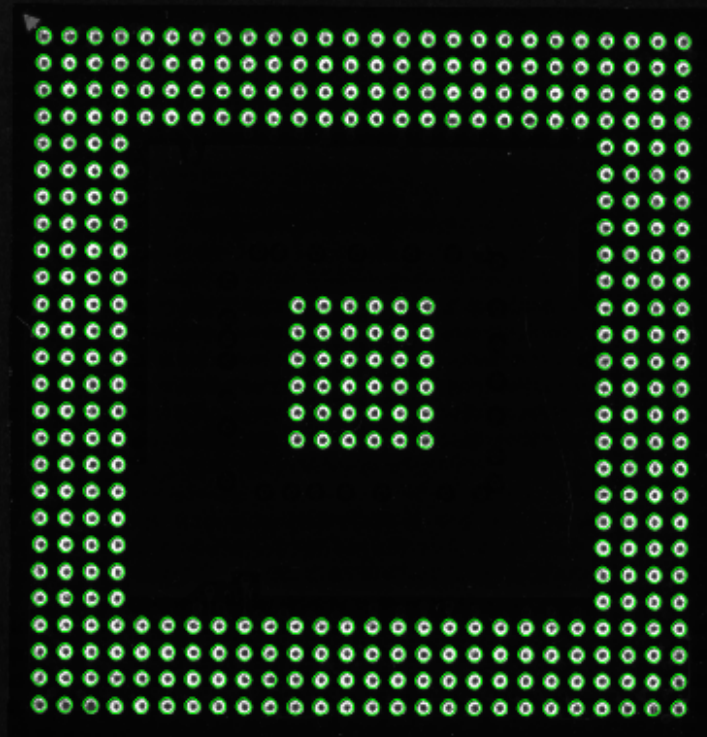
```

```

dev_set_color ('green')
dev_display (BallsSubPix)
*
* 6.1 错误放置的球（实际距离大于MaxDistance）设为青色
dev_set_color ('cyan')
for I := 0 to |Distances| - 1 by 1
    if (Distances[I] > MaxDistance)
        select_obj (BallsSubPix, BallWrongPos, IndexImage[I] + 1)
        dev_display (BallWrongPos)
        dev_display (BallWrongPos)
    endif
endfor
BallRad := sqrt((MinArea + MaxArea) / 2.0 / 3.1415926)
for I := 0 to BallsPerRow * BallsPerCol - 1 by 1
    if (BallMatrixRef[I] >= 0 and BallMatrix[I] < 0)
        *
        * 6.2 丢失的球设置为黄色。
        dev_set_color ('yellow')
        affine_trans_point_2d (HomMat2DInvert,
BallsRowsRef[BallMatrixRef[I]], BallsColsRef[BallMatrixRef[I]], RowImage,
ColImage)
        gen_ellipse_contour_xld (BallMissing, RowImage, ColImage, 0, BallRad,
BallRad, 0, rad(360), 'positive', 1.5)
        dev_display (BallMissing)
    endif
    if (BallMatrixRef[I] < 0 and BallMatrix[I] >= 0)
        *
        * 6.3 多余的球设置为红色。
        dev_set_color ('red')
        select_obj (BallsSubPix, BallExtra, BallMatrix[I] + 1)
        dev_display (BallExtra)
    endif
endfor
*
* 6.4 大小错误的球设置为蓝色。
dev_set_color ('blue')
dev_display (WrongAreaBallsSubPix)
*
* 6.5 形状错误的球设置为品红色。
dev_set_color ('magenta')
dev_display (WrongAnisometryBallsSubPix)
dev_disp_text ('Press Run (F5) to continue', 'window', 'bottom', 'right',
'black', [], [])
stop ()
endfor
disp_end_of_program_message (WindowHandle, 'black', 'true')

```

运行结果截图



Press Run (F5) to continue

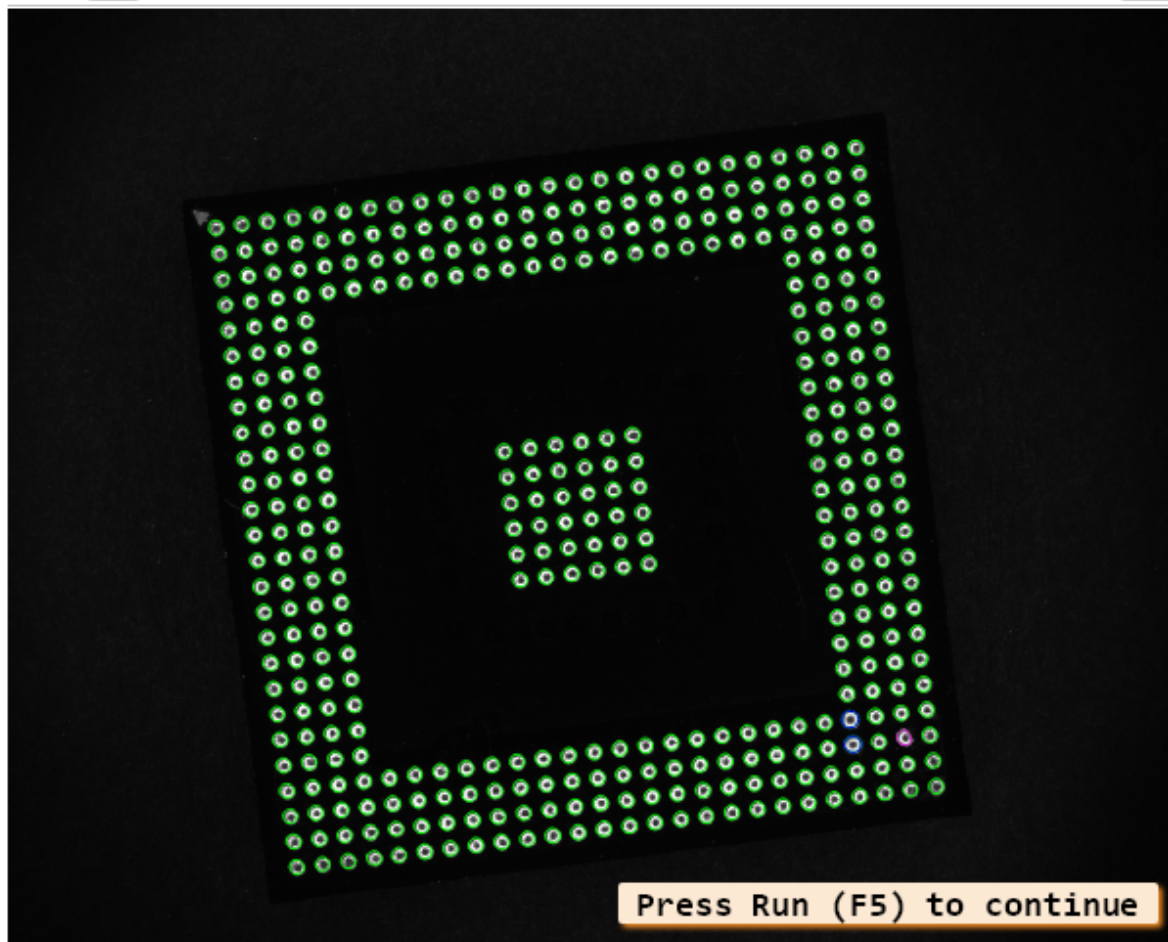
图形窗口: [H1FDF07FB4B0]



适应窗口



100 %



Press Run (F5) to continue

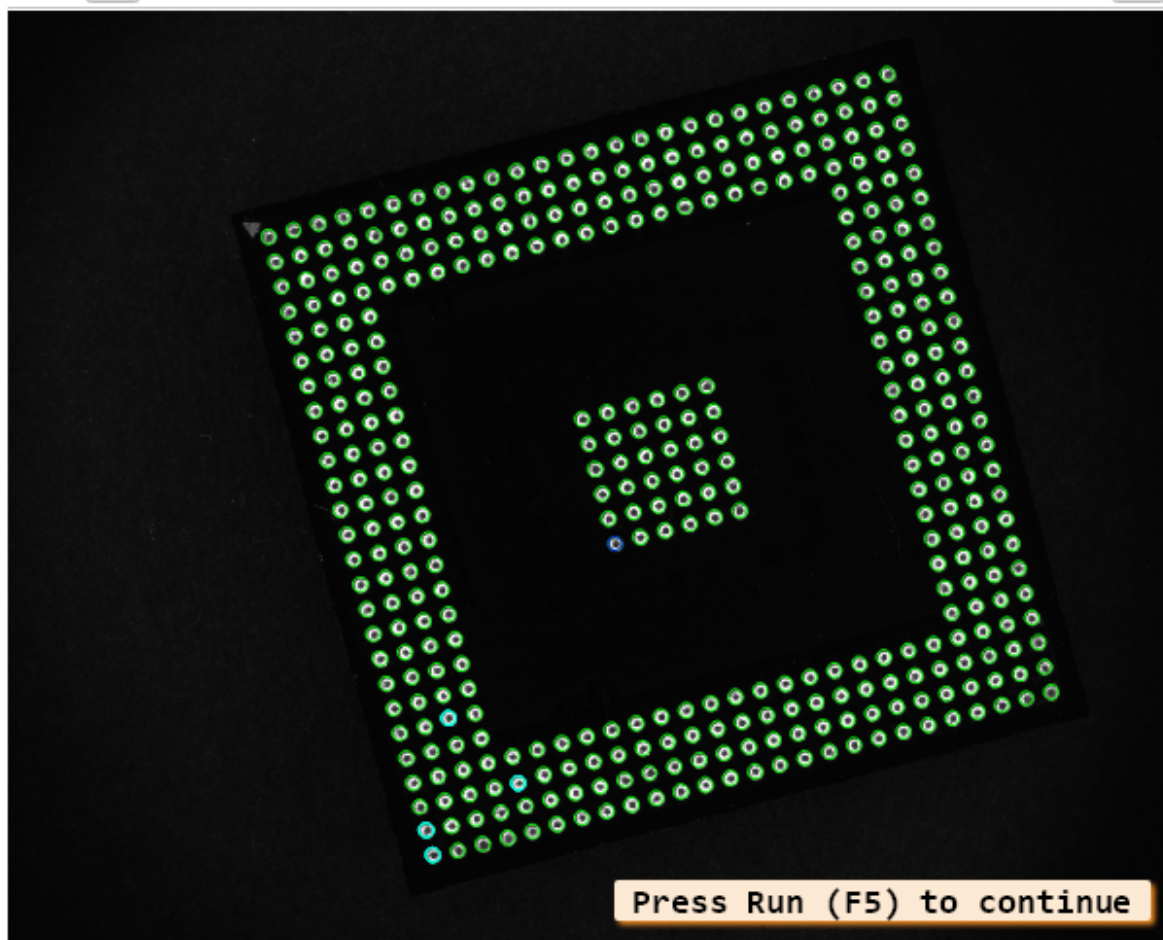
图形窗口: [H1FDF07FB4B0]

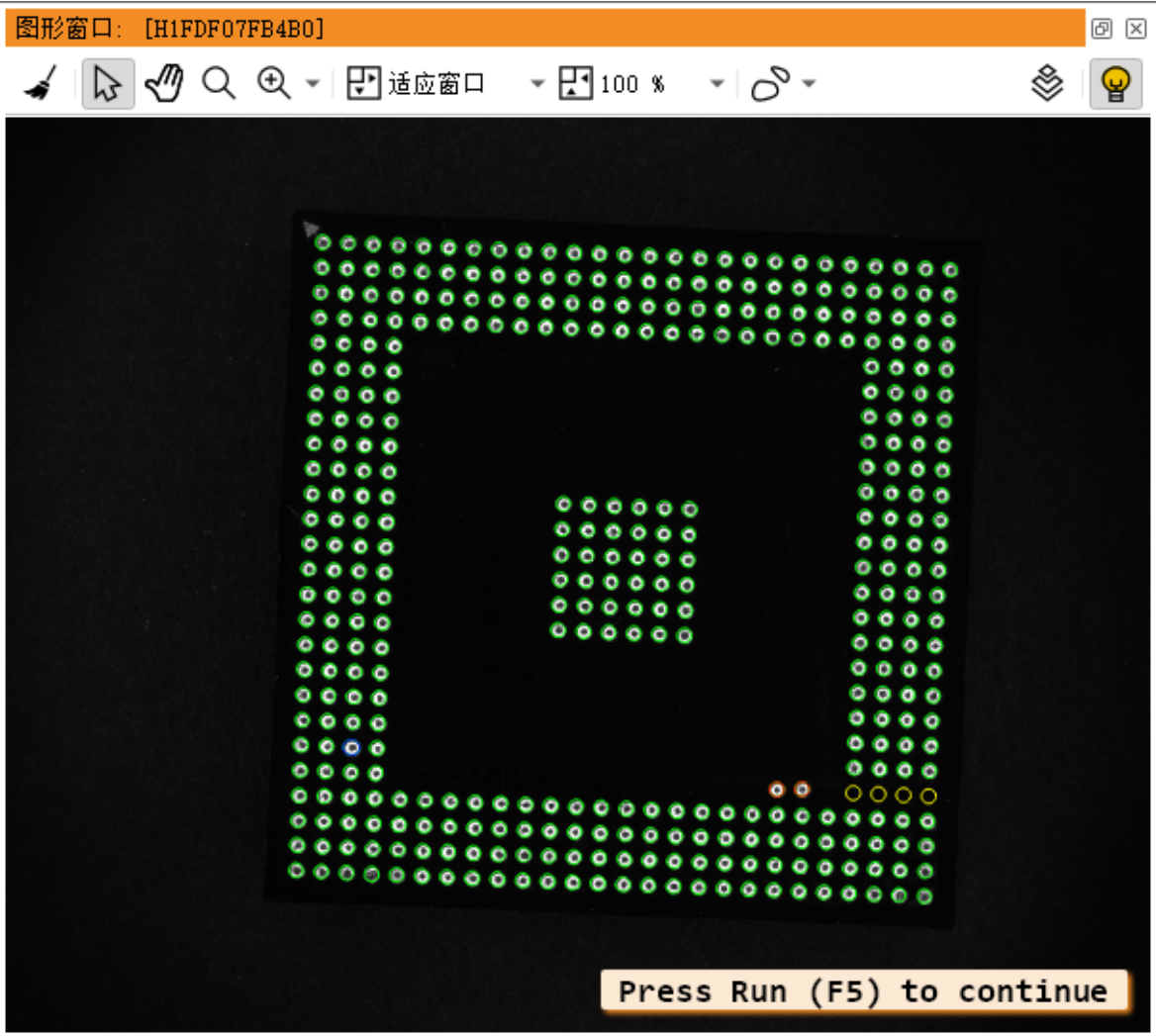


适应窗口



100 %





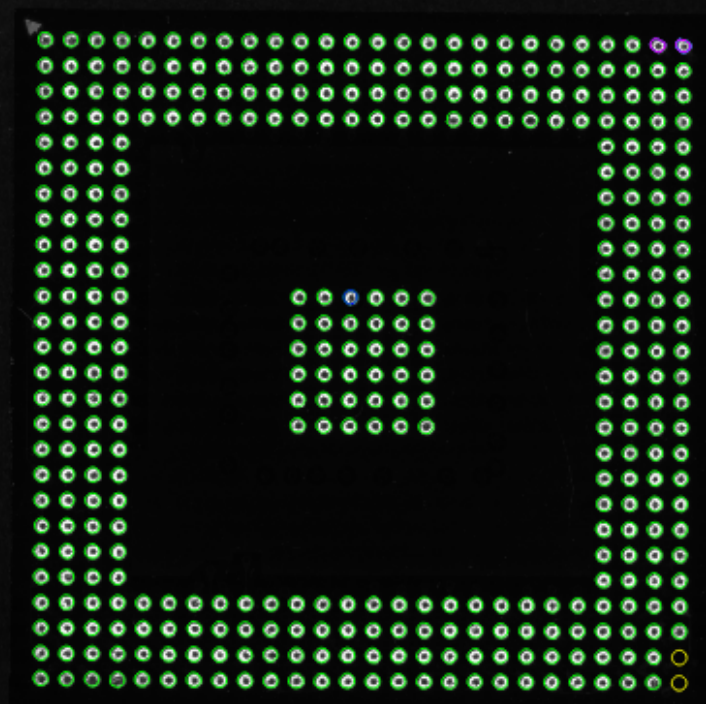
图形窗口: [H1FDF07FB4B0]



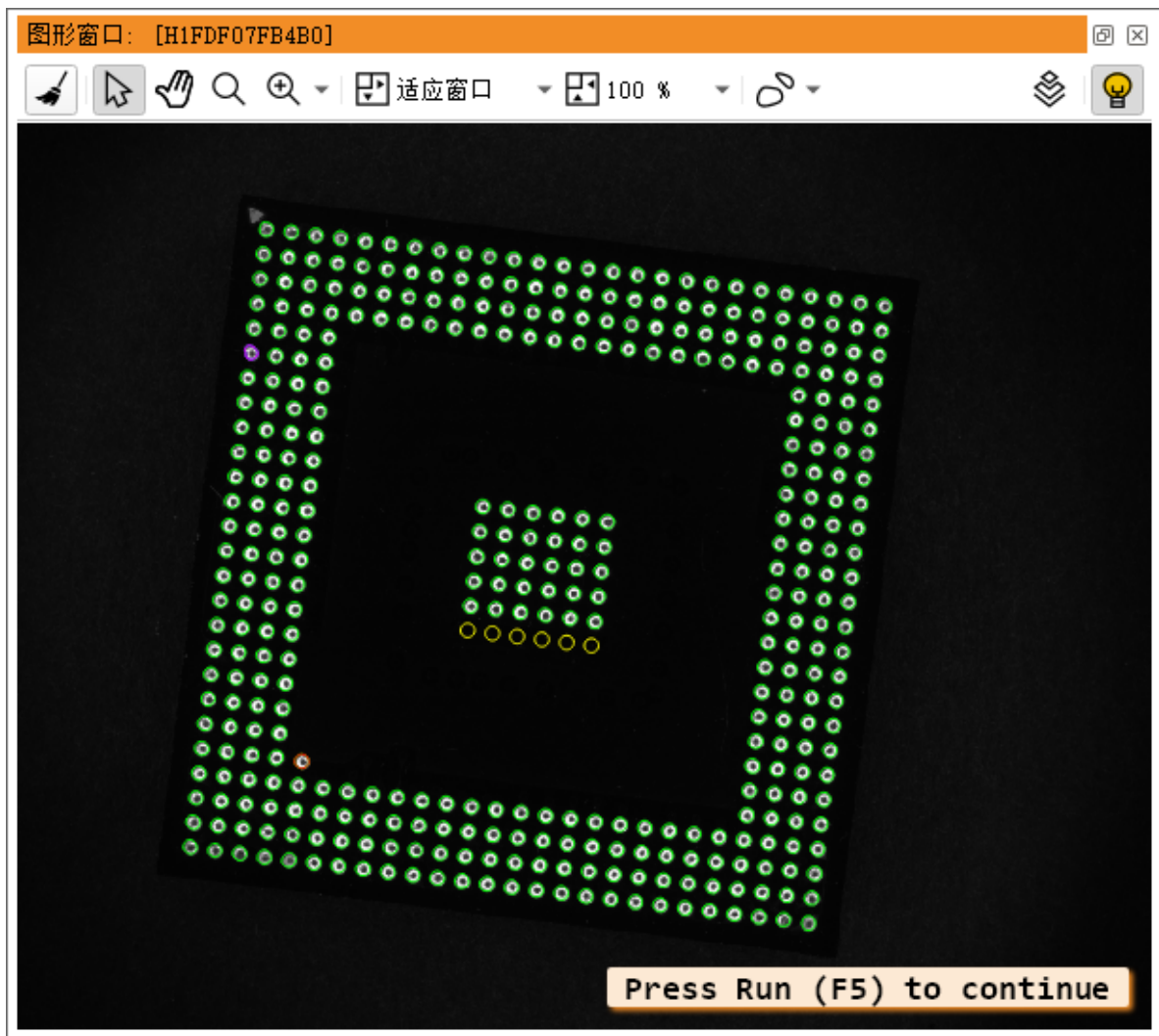
适应窗口



100 %



Press Run (F5) to continue



结果分析

图片序号	分析
1	完全正确，不存在异常
2	两个球大小错误，一个球形状错误
3	一个球大小错误，四个球位置错误
4	一个球大小错误，两个球多余，四个球缺失
5	一个球大小错误，两个球形状错误，两个球缺失
6	一个球形状错误，一个球多余，六个球缺失