

1	2	3	4	5	5
6	7	8	9	5	5
11	12	13	14	5	5

1	2	3	4	6	7
8	9	11	12	13	14
5	5	5	5	5	5

resin array

start of life

1	2	3	4	m	m	5	6	7	8	m	m	9	10	11	12	m	m
---	---	---	---	---	---	---	---	---	---	---	---	---	----	----	----	---	---

1	2	3	4	6	7
8	9	11	12	13	14
5	5	5	5	5	5

```

#include <opencv2/core/core.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/highgui/highgui.hpp>

using namespace cv;

int main() {
    Mat img = imread("img.png");
    if (!img.data) {
        cout << "Error loading image" << endl;
        return -1;
    }

    // Create a binary mask for the resin array
    Mat mask;
    cvtColor(img, mask, CV_BGR2GRAY);
    threshold(mask, mask, 128, 255, THRESH_BINARY);

    // Find contours of the resin array
    vector<Vec3i> contours;
    findContours(mask, contours, RETR_LIST, CV_RETR_CCOMP);

    // Sort contours by area
    sort(contours.begin(), contours.end(), [](Vec3i a, Vec3i b) {
        return a.area() > b.area();
    });

    // Get the largest contour (the resin array)
    Vec3i largestContour = contours[0];
    Rect resinArray(largestContour.x, largestContour.y, largestContour.x2, largestContour.y2);

    // Crop the resin array from the image
    Mat resinArrayCrop = img(resinArray);

    // Display the resin array
    imshow("Resin Array", resinArrayCrop);

    // Wait for a key press
    waitKey();

    return 0;
}

```



```

// ... (Additional code snippets related to the image processing)

// ... (Additional code snippets related to the image processing)

// ... (Additional code snippets related to the image processing)

```