Function Documentation: `eliminate_duplicate_circles`

1 Description

The `eliminate_duplicate_circles` function removes duplicate circles from a list based on their centers. It uses a distance threshold to determine if two circles are considered duplicates. Only unique circles are retained, and their indices are returned.

2 Function Definition

```python
def eliminate_duplicate_circles(circles, center_threshold):
    unique_indices = []

    for index, circle in enumerate(circles):
        x, y, r = circle
        duplicate_found = False

        for index2 in unique_indices:
            ux, uy, ur = circles[index2]
            distance = np.sqrt((ux - x)**2 + (uy - y)**2)

            if distance < center_threshold:
                duplicate_found = True
                break

        if not duplicate_found:
            unique_indices.append(index)

    return np.array(unique_indices, dtype=int)
```

3 Function Explanation

3.1 Step-by-Step Breakdown

Function 1: Initialize Unique Indices

Initialize an empty list to store indices of unique circles.

```python
unique_indices = []
```

Explanation: An empty list `unique_indices` is created to keep track of the indices of circles that are identified as unique. This list will be used to filter out duplicate circles.

Function 2: Iterate Through Circles

Iterate through each circle and determine if it is a duplicate.

```
for index, circle in enumerate(circles):
    x, y, r = circle
    duplicate_found = False
```

Explanation: The function loops through each circle in the `circles` list. For each circle, it initializes a flag `duplicate_found` to `False` to check if the circle is a duplicate.

Function 3: Check for Duplicates

Compare the current circle with previously found unique circles to check for duplicates.

```
for index2 in unique_indices:
    ux, uy, ur = circles[index2]
    distance = np.sqrt((ux - x)**2 + (uy - y)**2)

    if distance < center_threshold:
        duplicate_found = True
        break
```

Explanation: The function compares the current circle's center with those of previously identified unique circles. It calculates the distance between centers and checks if it is less than the `center_threshold`. If a duplicate is found, the `duplicate_found` flag is set to `True` and the loop exits.

Function 4: Store Unique Circle

If the circle is not a duplicate, add its index to the `unique_indices` list.

```
if not duplicate_found:
    unique_indices.append(index)
```

Explanation: If no duplicates are found for the current circle, its index is added to the `unique_indices` list, marking it as unique.

Function 5: Return Unique Circle Indices

Return an array of indices for the unique circles.

```
return np.array(unique_indices, dtype=int)
```

Explanation: The function converts the `unique_indices` list to a NumPy array of integer type and returns it. This array contains the indices of the circles that are considered unique.

4 Conclusion

The `eliminate_duplicate_circles` function is designed to identify and remove duplicate circles based on their centers. It ensures that only unique circles are retained and provides their indices in the output array.