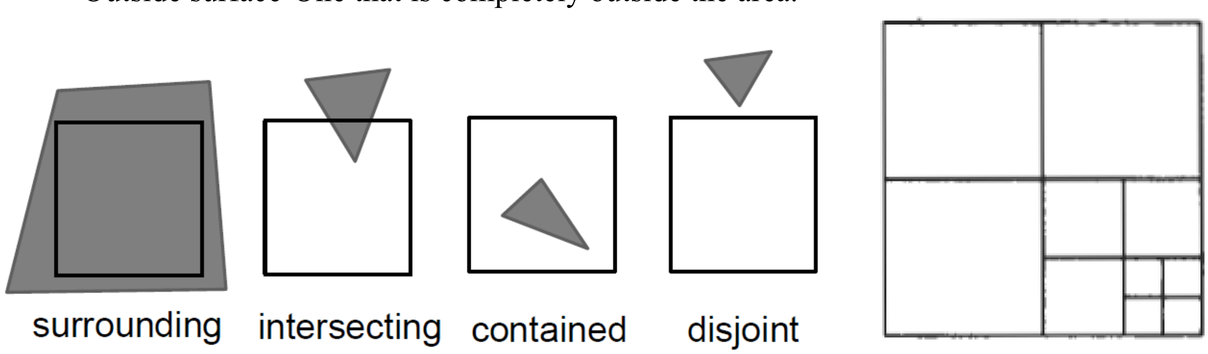
AREA-SUBDIVISION METHODS
This technique for hidden-surface removal is essentially an image-space method, but object-space operations can be used to accomplish depth ordering of surfaces.
The area-subdivision method takes advantage of area coherence (consistency) in a scene by locating those view areas that represent part of a single surface.
This method is applied successively by **dividing the total viewing area into smaller and smaller rectangles** until each small area is the projection of part of ***n*** single visible surface or no surface at all.
To implement this method, tests need to establish that can quickly identify the area as part of a single surface or show that the area is too complex to analyze easily.
Starting with the total view, apply the tests to determine whether subdivision of the total area must be done into smaller rectangles.
If the tests indicate that the view is sufficiently complex, subdivide it. Next, **apply the tests to each of the smaller areas**, subdividing these if the tests indicate that visibility of a single surface is still uncertain.
Continue this process until the **subdivisions are easily analyzed as belonging to a single surface** or until they are reduced to the size of a single pixel.
An easy way to do this is to successively divide the area into four equal parts at each step.
This approach is similar to that used in constructing a quadtree. **A** viewing area with a resolution of 1024 by 1024 could be subdivided ten times in this way before a subarea is reduced to a pint.
Tests to determine the visibility of a single surface within a specified area are made by comparing surfaces to the boundary of the area.
There **are** four possible relationships that a surface can have with a specified area boundary.
Relative surface characteristics:

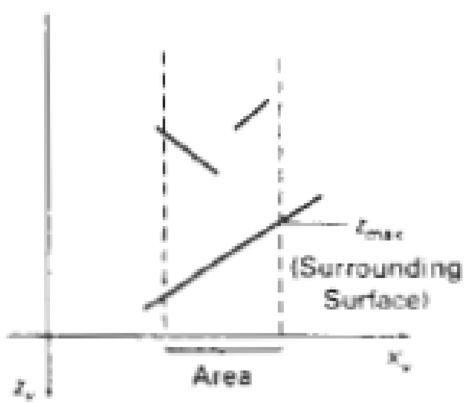
- Surrounding surface-One that completely encloses the area.
- Overlapping surface-One that is partly inside and partly outside the area.
- Inside surface-One that is completely inside the area.
- Outside surface-One that is completely outside the area.



The tests for determining surface visibility within an area can be stated in terms of these four classifications. No further subdivisions of a specified area are needed if one of the following conditions is true:
1. All surfaces are outside surfaces with respect to the area.
2. Only one inside, overlapping, or surrounding surface is in the area.
3. A surrounding surface obscures all other surfaces within the area boundaries.
Test 1 **can** be carried out by checking the bounding rectangles of all surfaces against the area boundaries.

Test 2 **can** also **use** the bounding rectangles in the **xy** plane to identify an inside surface.
For other **types** of surfaces, the bounding rectangles can be **used** as an initial check. If a single bounding rectangle intersects the area in some way, additional checks are used to determine whether the surface is surrounding, overlapping, or outside.
Once a single inside, overlapping, or surrounding surface has been identified, its pixel intensities are transferred to the appropriate area within the frame buffer.

One method for implementing test 3 is to order surfaces according to their minimum depth from the view plane. For each surrounding surface, we then compute the maximum depth within the area under consideration. If the maximum depth of one of these surrounding surfaces is closer to the view plane than the minimum depth of all other surfaces within the area, test 3 is satisfied.
Another method for carrying out test 3 that does not require depth sorting is to use plane equations to calculate depth values at the four vertices of the area for all surrounding, overlapping, and inside surfaces, If the calculated depths for one of the surrounding surfaces is less than the calculated depths for all other surfaces, test 3 is true. Then the area can be filled with the intensity values of the surrounding surface.
For some situations, both methods of implementing test 3 will fail to identify correctly a surrounding surface that obscures all the other surfaces. Further testing could be carried out to identify the single surface that covers the area, but it is faster to subdivide the area than to continue with more complex testing.
Once outside and surrounding surfaces have been identified for an area, they will remain outside and surrounding surfaces for all subdivisions of the area.
Furthermore, some inside and overlapping surfaces can be expected to be eliminated as the subdivision process continues, so that the areas become easier to analyze.
In the limiting case, when a subdivision the size of a pixel is produced, we simply calculate the depth of each relevant surface at that point and transfer the intensity of the nearest surface to the frame buffer.



Algorithm:

− Try to make an easy decision about which polygon is visible in a section of the image. If a decision cannot be made, subdivide the area recursively until one can be made.

− Work at image precision for subdivision, and at object precision for depth comparison

```
For each area do the following tests
1. are all polygons disjoint from area?
     if yes, display background color
2. only one intersecting or contained polygon.
     if yes, fill with background color, and then draw contained
     polygon or intersecting portion
3. one single surrounding polygon, no intersecting or contained
   polygons.
     if yes, draw area with that polygon's color
4. more than one polygon is intersecting, contained in, or
   surrounding, but only one polygon is  surrounding the area and is
   in front of others
     if yes draw area with that front polygon's color
5. otherwise, subdivide the area into 4 equal areas and recurse
```