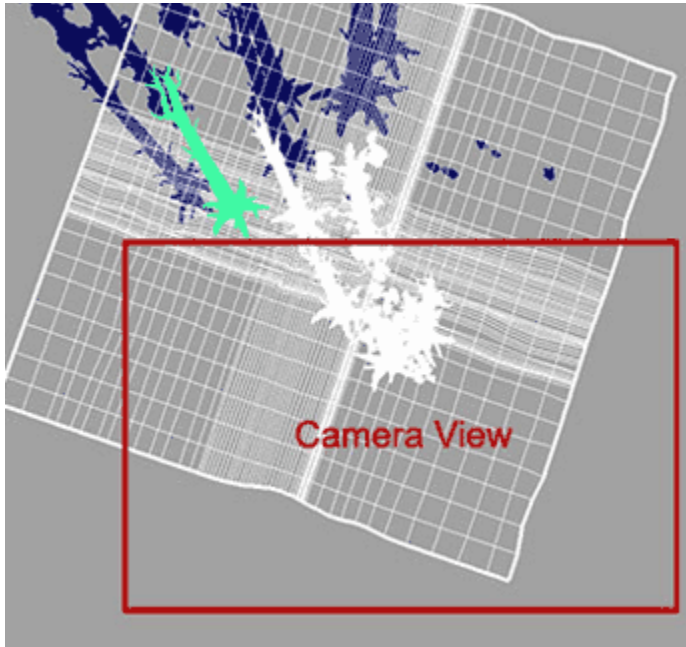


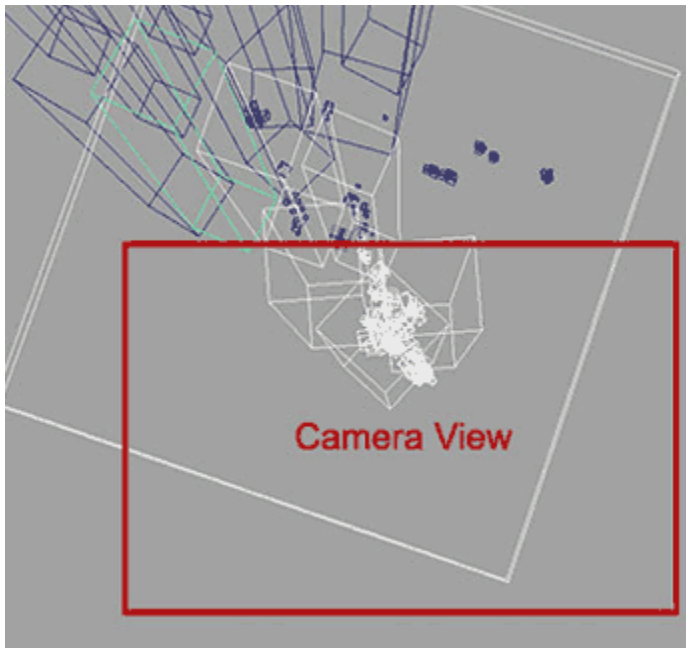
View Frustum Culling

(Maya C++ API)

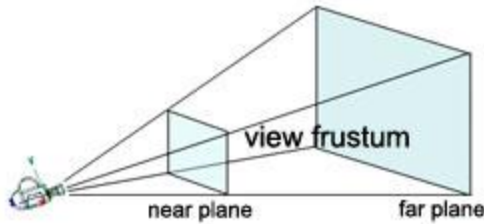
- Returns: string[] array of objects inside view frustum of <camera_name>



Wireframe View



Bounding Box View



The algorithm uses matrix multiplication on each object's bounding box to transform its endpoints into the local space of the camera and determine if any portion of the bounding box lies inside the camera frustum (which is bounded on either end by the near and far clip planes). The objects highlighted above were selected using `select -r `dkbObjectsInCameraView cameraName``. When switching to bounding box mode, we can see that every object whose bounding box is even partially in-frame is selected.

When processing nodes, the DAG hierarchy is descended only as deeply as necessary; if a parent object is completely contained within the camera view, its children will not be processed. ([hide pseudocode](#))

```
function Process(node):
    get bounding box of node
    if bounding box is completely outside frustum:
        do nothing
    else if bounding box is completely inside frustum:
        add node and all descendants to ListOfObjectsInFrustum
    else: // bounding box intersects frustum
        if node is a leaf node:
            add node to ListOfObjectsInFrustum
        else:
            for each child of node:
                Process(child)

for each top-level DAG node:
    Process(DAG node)
return ListOfObjectsInFrustum
```