```
struct Vertex2D { float_2 Pos; };
IUnknown * my_rotate(ID3D11Device* d3ddevice, float THETA,
                     int num_elements, const float_2 * data)
{
    // copy data into a DX buffer
    accelerator_view acc = create_accelerator_view(d3ddevice);
    array<Vertex2D,1> vertices(num_elements, data, acc);
    parallel_for_each(vertices.extent,
                      [=, &vertices] (index<1> idx) restrict(amp) {
        // Rotate the vertex by angle THETA
        float_2 pos = vertices[idx].Pos;
        vertices[idx].Pos.y = pos.y * cos(THETA) - pos.x * sin(THETA);
        vertices[idx].Pos.x = pos.y * sin(THETA) + pos.x * cos(THETA);
    });
    // return the DX buffer use of transformed data.
    return get_buffer(vertices);
}
```