Subject: Re: Tessellate Question--fewest convex polys
Posted by David Fanning on Fri, 08 Feb 2002 16:44:25 GMT

View Forum Message <> Reply to Message

Noname (parrhasius@altavista.com) writes:

> What's the best way to tessellate a polygon so that the resulting
> connectivity array contains the _fewest_ number of convex polygons instead
> of every single triangle? I don't want to cut up shapes like squares and
> octagons, only shapes with concavities or holes like lima beans and donuts.
> IDLgrTESSELLATOR always tessellates polygons into triangles, as I understand
> it; squares become 2 triangles, octagons become 6 triangles, etc...
>
> Using UNIQ and maybe MESH_MERGE I can get what I want by playing around with
> such a connectivity array of triangles, but as I am completely new to this I
> wonder if IDL already has this functionality or if this is a common and
> proven algorithm--or if I am just missing something obvious (most likely).
> Incidentally, what is the name for this type of tessellation? I know I've
> seen it implemented before in things like 3D video game level designers.
> For now I'd be happy with something that just worked in 2D.

I think what you are looking for is MESH_DECIMATE.

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Tessellate Question--fewest convex polys
Posted by Noname[1] on Fri, 08 Feb 2002 16:57:03 GMT

View Forum Message <> Reply to Message

I already tried MESH_DECIMATE and I'm not sure it does what I want. I tried to use it to decimate a square. Since a square is already a convex polygon, the function I need would do nothing to it, just return a connectivity array of 5 elements: [4, 0, 1, 2, 3]. MESH_DECIMATE either returned exactly the same thing as the IDLgrTESSELLATOR, a two triangle connectivity array (something like [3,0,1,2,3,0,2,3]) or else with a different polygon_percent it just returned one of these triangles, or half the square. This was the only time I tried to use it, but based on the results on the simplest case I concluded, probably incorrectly and prematurely, that it would not work. I can't even test it now because it will take me over an hour to get to a

computer with IDL, but just so that I might have a response waiting when I
get to that computer: Was I doing something wrong?   If MESH_DECIMATE really
does what I need, I apologize for this thread and don't bother to answer,
unless there's some trick to using it... I thought I checked it out.
Thanks.

"David Fanning" <david@dfanning.com> wrote in message
news:MPG.16cdae59743c72f19897f9@news.frii.com...
> Noname (parrhasius@altavista.com) writes:
>
>> What's the best way to tessellate a polygon so that the resulting
>> connectivity array contains the _fewest_ number of convex polygons
instead
>> of every single triangle? I don't want to cut up shapes like squares
and
>> octagons, only shapes with concavities or holes like lima beans and
donuts.
>> IDLgrTESSELLATOR always tessellates polygons into triangles, as I
understand
>> it; squares become 2 triangles, octagons become 6 triangles, etc...
>>
>> Using UNIQ and maybe MESH_MERGE I can get what I want by playing around
with
>> such a connectivity array of triangles, but as I am completely new to
this I
>> wonder if IDL already has this functionality or if this is a common and
>> proven algorithm--or if I am just missing something obvious (most
likely).
>> Incidentally, what is the name for this type of tessellation? I know
I've
>> seen it implemented before in things like 3D video game level designers.
>> For now I'd be happy with something that just worked in 2D.
>
> I think what you are looking for is MESH_DECIMATE.
>
> Cheers,
>
> David
> --
> David W. Fanning, Ph.D.
> Fanning Software Consulting
> Phone: 970-221-0438, E-mail: david@dfanning.com
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Tessellate Question--fewest convex polys
Posted by David Fanning on Fri, 08 Feb 2002 18:17:42 GMT
View Forum Message <> Reply to Message

Noname (parrhasius@altavista.com) writes:

> If MESH_DECIMATE really
> does what I need, I apologize for this thread and don't bother to answer,
> unless there's some trick to using it... I thought I checked it out.

I've never had the ...er, opportunity to use this
yet. I'm afraid someone else will have to address
this question.

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Tessellate Question--fewest convex polys
Posted by Pavel A. Romashkin on Fri, 08 Feb 2002 18:39:56 GMT
View Forum Message <> Reply to Message

I am veering off the question.
I had limited success with the Tessellator object. Returned connectivity
arrays were not convex when plotted.
Pavel

Noname wrote:
>
> Hello all,
>
> What's the best way to tessellate a polygon so that the resulting
> connectivity array contains the _fewest_ number of convex polygons instead
> of every single triangle? I don't want to cut up shapes like squares and
> octagons, only shapes with concavities or holes like lima beans and donuts.
> IDLgrTESSELLATOR always tessellates polygons into triangles, as I understand
> it; squares become 2 triangles, octagons become 6 triangles, etc...
>
> Using UNIQ and maybe MESH_MERGE I can get what I want by playing around with
> such a connectivity array of triangles, but as I am completely new to this I
> wonder if IDL already has this functionality or if this is a common and

> proven algorithm--or if I am just missing something obvious (most likely).
> Incidentally, what is the name for this type of tessellation? I know I've
> seen it implemented before in things like 3D video game level designers.
> For now I'd be happy with something that just worked in 2D.
>
> Thanks for any help.

Subject: Re: Tessellate Question--fewest convex polys
Posted by karl_schultz on Fri, 08 Feb 2002 23:11:23 GMT
View Forum Message <> Reply to Message

David Fanning <david@dfanning.com> wrote in message
news:<MPG.16cdc43310dbf739897fb@news.frii.com>...
> Noname (parrhasius@altavista.com) writes:
>
>> If MESH_DECIMATE really
>> does what I need, I apologize for this thread and don't bother to answer,
>> unless there's some trick to using it... I thought I checked it out.
>
> I've never had the ...er, opportunity to use this
> yet. I'm afraid someone else will have to address
> this question.
>
> Cheers,
>
> David

MESH_DECIMATE is used to simplify polygonal meshes by collapsing mesh vertex pairs, selecting the pairs carefully so that the simplified mesh represents the original mesh as closely as possible.

It is typically used for things like 3D terrain maps and polyognal models in order to produce a mesh that has fewer triangles, yet still retains the original 3D shape as much as possible. Try the little demo program called 'decimate' in the IDL 5.4 and 5.5 releases.

Using MESH_DECIMATE on a 2D (flat) mesh is a little out of the ordinary, since the criteria used to select the vertex pair to collapse would select any vertex pair in the mesh. That is, it does not matter which pair it selects; you will always have a flat mesh regardless of which and how many pairs are collapsed. This would make the pair collapse look a little random, or related to the order of the triangles in the mesh. But it would still reduce the number of triangles in the mesh, depending on the percentage setting.

It is a little hard to say if this is the right tool for the data described in the original post. MESH_DECIMATE starts by turning the

mesh into triangles, so it actually increases the number of polygons
in the mesh as it gets started. This is required, because the vertex
collapse criteria evaluation depends on the mesh only containing
triangles. Then, it will collapse vertices and the number of tris
will decrease and the size of the triangles will increase. You'll end
up with a bunch of largeish triangles that define your mesh.

If you (original poster) really needs to have large n-gons instead of
large triangles, then this won't work. I'd suggest googling around on
the net for this sort of thing. In a way, this is sort of a reverse
constrained tessellation - combining triangles into convex polygons.
I don't think that there is anything in IDL that will do this
directly. The main purpose of the tessellator object is to allow you
to convert possibly concave polygons into a set of triangles that can
be drawn using IDLgrPolygon.

Karl
RSI

Subject: Re: Tessellate Question--fewest convex polys
Posted by Mark Hadfield on Sun, 10 Feb 2002 21:15:22 GMT
View Forum Message <> Reply to Message

"Pavel A. Romashkin" <pavel_romashkin@hotmail.com> wrote in message
news:3C641B7B.F4FA5699@hotmail.com...

> I am veering off the question.
> I had limited success with the Tessellator object. Returned
> connectivity arrays were not convex when plotted.

Really? That is not my experience. The tessellator always produces
triangles, which are convex by definition.

My gripes with the tessellator are:

1. It takes a long time for large datasets

2. It is completely intolerant of badly formed data
(eg. intersecting segments)

3. It tests the area of the triangles it generates and reports an
error ("colinear vertices") if this area is less than 0.5E-6. There
are situations where this is inappropriate. The work-around is to
multiply the input data by a constant before tessellating.

Mark Hadfield
m.hadfield@niwa.co.nz http://katipo.niwa.co.nz/~hadfield