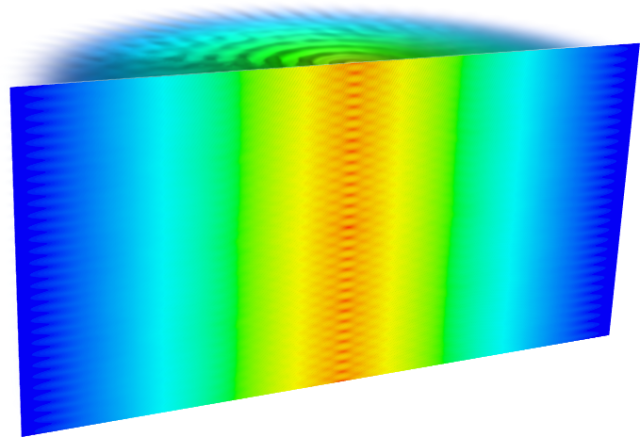
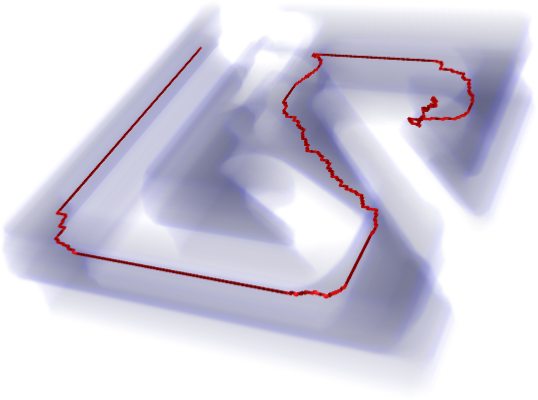
**[New paper out!](https://0fps.net/2010/06/01/new-paper-out/)**

Posted on [June 1, 2010](https://0fps.net/2010/06/01/new-paper-out/)by [mikolalysenko](https://0fps.net/author/mikolalysenko/" \o "View all posts by mikolalysenko)

Haven’t updated this blog in a long time, but I figure that since it is summer now maybe I will finally be able to keep this thing regularly updated (see how long that lasts…)  Anyway, some good news is that my latest paper has been accepted to the SIAM/SIGGRAPH conference on solid and physical modeling.  Here is a link to the download:

<http://sal-cnc.me.wisc.edu/index.php?option=com_remository&Itemid=143&func=fileinfo&id=183>

[](https://0fps.files.wordpress.com/2010/06/gear_cutaway.png)

[](https://0fps.files.wordpress.com/2010/06/path_vol.png)

There’s a bunch of neat ideas in here, but the big idea here is the Minkowski product.  The motivation for this comes from the basic Minkowski sum.  If we recall, for two sets A, B \subseteq \mathbb R^n, their*Minkowski sum*is defined as:

A \oplus B = \{ a + b | a \in A, b \in B \}

Or alternatively,

A \oplus B = \bigcup \limits_{a \in A} a + B 

The idea behind the Minkowski product is to replace \mathbb R^n with an arbitrary group, G.  If we do this, then we can define the*Minkowski product* over G in the following way:

A \stackrel{G}{\otimes} B = \bigcup \limits_{a \in A} a B 

Much like how the Minkowski sum is useful for collision detection of translating objects, the generalized Minkowski product can be used for collision detection between translating and rotating bodies.  Similarly, we can also define a Minkowski quotient which is analogous to the Minkowski difference:

A \stackrel{G}{\oslash} B = \bigcap \limits_{a \in A} a B 

And even better yet, we show how to compute these quantities using convolution algebras!  These operators turn out to be very useful in understanding things like partial symmetries of solids, mechanism workspaces and robotics.  All of the gory details and more are in the paper!