



# GeoHipster

[Home](#) [News](#) [Interviewees](#) [Calendar and Merch](#) [About](#) [Posts](#)

[Contact](#)

## Howard Butler: "Like a good song, open source software has the chance to be immortal"



Howard Butler

*Howard Butler attended Iowa State University and departed with Bachelor's and Master's degrees after studying parts of Agronomy, Agricultural Technology, and Agricultural Engineering. He learned GIS software development during his thesis effort, where he needed to make ArcView 3.x do a complicated and completely unrealistic analysis. After failing*

*to find a precision agriculture job because a GPS for a tractor cost \$2,000 at the time, the Iowa State [Center for Survey Statistics and Methodology](#) took a chance on him to develop some GIS data collection and management software for the [National Resources Inventory](#). Fifteen years later he's helping to write open source software that's powering data management systems for autonomous vehicles.*

*Howard lives in Iowa City, Iowa with his wife Rhonda, his two boys Tommy and Jack, two dumb cats, and a squirrel or raccoon or something that takes residence in his attic every winter despite efforts otherwise. He has a neglected blog at <https://>*

[howardbutler.com/](https://howardbutler.com/), and he tweets less and less at <https://twitter.com/howardbutler>.

*Howard was interviewed for GeoHipster by Randal Hale.*

## Q: Howard – where are you located and what do you do?

A: My three-person company called [Hobu, Inc.](#) is located in Iowa City, Iowa, and we write, manage, and enhance open source point cloud software and help our clients use that software to solve their challenging problems. We initially focused on LiDAR (Light Detection and Ranging — think radar with lasers) with a project called [libLAS](#), evolved that into [PDAL](#) (GDAL for point clouds), and then continued with streaming technology in the form of [Greyhound](#) and [Entwine](#).

I started contributing to open source with [MapServer](#) and [GDAL](#) back in 2002 when I discovered it was the only software capable of building the systems my job demanded at the time. I came to enjoy the camaraderie and common purpose those good projects exuded, and I learned over the years how to contribute in a way that matched my skills. Among other things, that evolved into writing a number of geospatial Python bits (you can thank/hate me for plenty of [ogr.py](#) and [gdal.py](#)) and helping to author a bit of the [GeoJSON](#) specification (you can thank/hate me for coordinate systems there).

In 2007, I struck out on my own and promptly learned that I didn't know how to run a business. My banker still doesn't really understand how or why we give away our software, but people get it when I say our product is consulting with a software toolkit we incidentally give away. Over the years we've built up a stable client base that values what we do and how we do it, and I think that the software we've written will outlast my company or my career because it represents solutions to problems people hate to solve again and again.

## Q: So how did you end up working with

# LiDAR? I've had the chance to use PDAL and see some of your presentations at FOSS4G and FOSS4GNA.

A: The Iowa Department of Natural Resources led Iowa to be one of the first states to do a statewide LiDAR collection, and they had a grad student semester of funding they wanted to use to be able to use Python for **ASPRS LAS data** management, verification, and inspection. There was no open source requirement, but since it was what I was doing otherwise, it seemed natural to build a library that anyone could use. **Mateusz Loskot** and I started working on what became libLAS to achieve it, and once it was clear it was viable, I was able to attract more funding to enhance and improve it.

U.S. Army Corps of Engineers found libLAS and wanted to do a lot more with it — supporting a bunch more formats, getting it speaking to databases, and enhancing it to do a bunch more algorithmically. We learned stacking all those desires on a library based on the LAS format wasn't a great fit. We started **PDAL** (Point Data Abstraction Library — pronounce it the same way you do GDAL 😊) after some fits and starts and it has matured into a general-purpose library for building geospatial point cloud applications.

PDAL takes **GDAL's VRT** pipeline approach and puts it into the context of geospatial point clouds, but with JSON instead of XML. It works on Windows, OSX, and Linux, and it has a command line application like GDAL to drive processing. Its workflow is optimized to template data operations and batch them up over a pile of data with whatever batching/queuing/cloud tools you have. That might be GNU parallel if you want to melt your laptop locally or something like AWS SQS in a cloud situation.

## Q: I saw your presentation on Gerald Evenden at FOSS4G in Boston. Did he

# know that the PROJ library..or software... was going to go as far as it did? Actually – what does PROJ do?

A: PROJ or PROJ.4 is a cartographic reprojection library that was written by Gerry Evenden at USGS in the 1980s and 90s. It contains the math to reproject coordinates from UTM to Plate Carrée, for example. Gerry originally intended for PROJ to be a cartographic projection library (pure math only!), but in the 90s, Frank Warmerdam came along and started adding convenience for geodetic transformation (datum shifting). This caused some creative differences, but that geodetic convenience enabled PROJ to be bootstrapped or ported into almost every open source geospatial software package in some form or another.

While attempting to dig up some old documentation, I discovered Gerry died in 2016. This saddened me because I've felt that Gerry didn't get his due for the impact that PROJ has on the entire geospatial software ecosystem. It is truly everywhere — open source, commercial, and government software all depend upon PROJ. I submitted my FOSS4G 2017 talk in an attempt to tell his story and shine the spotlight on him even though he probably would have detested it.

I'm a fan of 60s and 70s rock n roll, and now that those guys are starting to die off, people are rediscovering a lot of back catalog. Plenty of it is still crap, but songs that shined then often sparkle today. Those songs were written for the audience of that time, but a good one can transport you there even if you weren't a part of it. Like a good song, open source software has the chance to be immortal. It is optimized for solving today's problem in today's context, but a few programs and libraries end up lasting multiple generations. Unlike a hit song, open source software isn't a static fount of royalties. It is a liability that must be maintained or it will crumble back into the ground. People need to cover it, make it their own, and feed it attention as Paul Ramsey wonderfully described in his FOSS4G 2017 keynote.

For PROJ, two longtime contributors have been covering the song and keeping the music alive. Thomas Knudsen and Kristian Evers from the Danish Agency for Data

**Supply and Efficiency** (kind of like Denmark USGS) refactored PROJ to be a full service geodetics library and they have modernized its API in the process. Kristian has led this **PROJ 5.0.0 release process**, and everyone's software is now going to be able to get a lot smarter about geodetic transformations. While the old APIs are still available so as to not break existing software, their improvements will make PROJ last for another couple of generations.

## Q: Thoughts on Mapzen shutting down?

A: Anxious and hopeful. As someone with an organization and employees, the thought of having to tell them they're now on their own is a front-of-mind fear. Organizations fail for many reasons despite the effort of the people pouring their sweat into them. To work so hard and have it be called a failure doesn't seem fair, but I'm thankful for Mapzen's postmortems, which have given everyone the chance to learn.

I'm hopeful due to the fact that I think Mapzen's employment model demonstrated a successful one for the employees. Many Mapzen'ers worked out in the open on public projects, and in the process made themselves and the teams they belonged to more valuable for it. Developing open source software in public, as opposed to never going out beyond your own wall, is something that makes you a better software developer. You have to listen to rightful criticism about your software, and you have to temper your emotional response to people rationally not liking the precious thing you just made (ok, not always). To solve hard problems in public leaves you exposed, but in exchange for that vulnerability, you generate a professional currency that follows you the rest of your career.

An influential quote I saw early in my career came from **Tim Peters of the Python language**:

You write a great program, regardless of language, by redoing it over & over & over & over, until your fingers bleed and your soul is drained. But if you tell newbies *that*, they might decide to go off and do something sensible, like bomb defusing<wink>.

The only thing you can do is make your software suck slightly less every day you

touch it. My formative experience in geospatial open source was watching folks like Frank Warmerdam, Steve Lime, Martin Davis, and Markus Neteler do exactly that. They controlled the complexity in front of them, resisted the urge to overdesign a solution, and they treated everyone with respect even when they didn't deserve it. I've tried to follow their approach with my projects, although I'd consider myself a worse developer than each of them by most measurements.

**Q: You were there at the beginning for OSGeo back in 2006(?) I think. How has it changed or remained the same? How did you get pulled into the organization?**

A: OSGeo's reason to exist in 2006 was different than it is in 2018. In 2006, it was supposed to be a group of geospatial software projects with a common thread about open source. In 2018, it is a group of people with a shared interest in open source geospatial software. The former was frustrating for different reasons than the latter, but it has been an organization that achieved substantial things despite the messy way in which it is able to go about it. Many of its challenges relate to the fact that it is a volunteer organization throughout, and personalities with drive and determination can have short-run impact, but long-run sustainment is very difficult. Recently, it has slowed its precession about the axis of outreach, education, and conferences, which are topics that fit the current makeup of the organization very well.

I've had many roles in the organization over the years, including helping to set up some of the first project infrastructure and acting as a board member. In 2006, software project infrastructure was a real cost, but in 2018, access to repositories, mailing lists, continuous integration, and bug reporting can all be had in exchange for some spam tolerance. Recently my contributions have been presenting at conferences and being a strong supporter of the mid-winter **OSGeo Code Sprint** that has oscillated back and forth between the EU and North America. Sprints are a primary opportunity for developer camaraderie and collaboration, and they provide the high-bandwidth communication forum for projects to grow and enhance each

other.

**Q: I'm not a developer by any stretch – but I like going to talks by developers on their software. You've built PDAL to manipulate LIDAR Data – what's the weirdest use case you've seen for PDAL so far?**

A: Nothing too weird, but it is an everyday occurrence for users to use the tools in ways we didn't foresee or intend. Every permutation of data size, composition, and fault gets hit eventually. For every success story using PDAL in a way we never thought of, there's a corresponding failure story due to assumptions that don't line up. Many times a great bug report is simply a challenge of those assumptions.

**Q: What's the Best Thing about Iowa?  
What's the Worst Thing? I drove through it once and didn't stop but long enough to eat a sandwich.**

A: The Public Land Survey System in Iowa means I'm never lost. You probably weren't ever concerned on your drive either. Also, the proximity to so much animal agriculture means that meat-as-a-condiment to more meat isn't just a specialty no-carb lifestyle choice here. You would think with 90+ percent of the land in the state used for agriculture there would be more vegetables around.

It's not the worst, but opportunities for Big Culture stuff like museums, art, and music shows are somewhat limited here, especially once you get out of the larger towns. Lack of diversity is a challenge too, although you find it in places in Iowa you wouldn't

expect. These are the same challenges for all rural states with aging, out-migrating populations.

## Q: Can you tell us something people might not know about you?

A: I grew up on a corn and soy farm in Southern Minnesota, and I was convinced that maps and computers were interesting after some quality time on the Dinty Moore Beef Stew assembly line. I have a pilot's license I haven't used in more than a decade, and my car was once struck by lightning while driving down the freeway at 70 mph (I shouldn't have bought it back from the insurance company). A long time ago, I won an Esri Conference award using AVPython and ArcView 3.x, and I could still sling VTables and FTables around in Avenue if I was cornered.

## Q: Almost 4 years ago we defined the geohipster to be a person who lives on the outskirts of mainstream GIS. Would you describe yourself as a geohipster?

A: I guess. GIS™ as a name is an outdated view of how the intersection of geography, computers, and databases is to be constructed. Each of its areas has been dumped over at least a couple times since GIS™ as a fashionable term came to describe our industry. Many still GIS™ on desktop software with a 2D map frame and 🔍 zoom and 🖱️ pan icons like twenty years ago, but geo+computers+databases is now oriented toward phones, sensors, and deriving locality from incidental data with cloud computing and pervasive networking. To call what's going on with all of that GIS™ seems rather trite.

## Q: I leave the last question to you –



# anything you want to tell the readers of GeoHipster?

A: Please make sure to buy a GeoHipster calendar or a t-shirt or something. We're all just learning here, and sites like this one make the job much easier and need our support.

---

Posted March 5, 2018 in [Geohipsters](#), [Interviews](#)  
by GeoHipster Team

Tags:

[howard butler](#)

[Previous](#)   [Next](#)

## Comments

2 responses to "Howard Butler: "Like a good song, open source software has the chance to be immortal""



Even Rouault

March 5, 2018

There's an error on the family name of one of the current proj contributors. It should read Thomas Knudsen (not Hansen)



Atanas Entchev

March 6, 2018

Corrected. Thank you for the feedback!

GeoHipster

**A bicycle ride around the world begins with a single pedal stroke – Scott Stoll**

