

# David LaRose's SCS Home Page




---

National Robotics Engineering Center  
 Carnegie Mellon University  
 email to: [dlr@cs.cmu.edu](mailto:dlr@cs.cmu.edu)  
 Phone: (412) 688-0918  
[PGP Public Key](#)

---

02135

## Work

I work at the [National Robotics Engineering Center](#) (NREC), which is part of Carnegie Mellon's [Robotics Institute](#). Our mission is to get working robotics & computer vision technology out in the world. Here are links to a few of the projects I've worked on:

- In the [UltraStrip](#) project, we developed a robot to safely and effectively remove toxic paint from the outside of ship hulls. This is an amazing piece of engineering. The robot sticks to the ship hull using magnets, and can even work upside down. The real movers and shakers on this project were [John Bares](#) and [Bill Ross](#). Last time I checked, this technology was in use at shipyards on three different continents!
- I've done quite a bit of work in [X-Ray/CT Registration](#) for medical applications. This project is an extension of my [thesis](#) work. We have a product-level suite of registration libraries, which is available for licensing.
- Recently, I've been working on a DOE funded project to develop [automated conveyor belt inspection](#) software. The BeltVision system is available for purchase from [Beitzel Corp.](#) Special thanks to [CONSOL Energy](#), who funded the initial development of this system, and who are partners on the DOE project.

I'll add links to other project pages as they become available.

## Free Software

Here are some links to software I've written:

- [dlr\\_libs](#) is a collection of libraries I initially wrote when I was consulting. I wanted a common platform of unencumbered code for use with different clients, so I wouldn't have to start from scratch each time. I use it as a foundation for much of my software. I've released it under a pretty permissive license already, but let me know if you want different terms.
- This isn't really software, but here's a [HOWTO](#) I wrote on setting up a simple network boot for stateless ([Debian](#)) GNU/Linux boxes. It's different from the average netboot setup in that the entire OS is in the network boot image. This is good because you don't have to have a separate NFS partition hanging around, and because it makes it easier to keep track of the state of the boxes you're booting. This is bad because it pretty severely limits how much stuff you can fit into the image.
- [xComplete](#) is an X11 client that watches what you type, politely suggests completions, and then fills in the ends of long (>5 characters) words and variable names. It's kind of like Dean Pomerleau's auto-completion.el package, except that it's not as cool, and it works with most X11 applications (not just emacs). I wrote this almost 10 years ago, using C++, [Python](#), and [SWIG](#), and I never made the installation work well. OTOH, I still use it every day. If you're not in a hurry,