Nate Sanders

Email Address: nate@natesanders.com
Mailing Address: 2644 Oahu Ave - Honolulu, HI 96822 - USA
Telephone: 859-421-1776

Summary

- MS in Computer Science with emphasis on Computer Vision, specifically Projector/Camera systems
- 12 years experience in C/C++/Perl and Linux/Unix systems programming
- Lead Developer on the prototype for the National Science Foundation's Ultra-Definition display standard at Mersive Technologies (http://www.mersive.com/)
- Significant experience in Linux/Unix systems programming including interprocess communication, threading, non-blocking I/O, multicast/unicast network programming, scatter/gather I/O, etc
- Familiar with non-mainstream programming languages (Lisp, Smalltalk, ML, others) and their concepts
- Significant experience in OpenCV/OpenGL/other image processing, computer vision, graphics libraries
- Experience as a Linux/Solaris system administrator
- Fast learner who quickly acquires domain-specific knowledge
- Strong mathematical background

Education

M.S., Computer Science, May 2007
University of Kentucky, Lexington, Kentucky USA
Advisor: Dr. Christopher Jaynes
The emphasis of my graduate work was computer vision/machine vision.

B.S., Computer Science, 1999
University of Kentucky, Lexington, Kentucky USA
Awarded a Singletary Scholarship, the University of Kentucky's most prestigious 4-year award.

Research and Career Interests

Computer vision, mathematics, machine learning, artificial intelligence, prototyping, automation, programming languages, systems research/design/implementation, cloud computing

Publications

1.Nathaniel Sanders and C. Jaynes. Class-specific Color Camera Calibration with Application to Object Recognition, WACV 2005, 2005.
http://www.natesanders.com/mypapers/Sanders-WACV2005.pdf

2.Christopher Jaynes, Amit Kale, Nathaniel Sanders, and E. Grossman. The Terrascope Dataset: A Scripted Multi-Camera Indoor Video Surveillance Dataset with Ground-truth, Proceedings of the IEEE Workshop on VS PETS, 2005.
http://www.natesanders.com/mypapers/jaynespets2005.pdf

3.Nathaniel Sanders and C. Jaynes. A Camera-Based Energy Relaxation Framework to Minimize Color Artifacts in a Projected Display, 2006 Conference on Computer Vision and Pattern Recognition Workshop, 2006.
http://www.natesanders.com/mypapers/procams2006.pdf

Professional Experience

August 2008 – December 2008. Contractor/Developer. Laboratory for Interactive Learning Technologies (LILT) at the University of Hawaii-Manoa, Honolulu, HI.
Consulted on a variety of issues regarding computer vision, augmented reality, and systems architecture – particularly involving face detection, face recognition, the evaluation of these and the integration of these into mobile/ubiquitous systems. Implemented both client and server pieces for a near-real-time smartphone-based video face-detection system. I wrote the client using C++ on the Nokia N95 running Symbian OS and the server using C++ running on Linux. In addition to the near-real-time system, I also integrated face detection as a RESTful Ruby on Rails service for SocialSense, a mobile phone-based social networking application.

February 2008 – July 2008. Contractor/Developer. Mersive Technologies, Lexington, KY.
As lead software architect/developer on the Linux side of Mersive's upcoming Stingray platform, I designed, implemented, managed, modified, and combined large code bases involving Xlib, the X11 protocol, Xdmx, and OpenGL to create a cohesive Ultra Definition (considerably higher resolution than standard HD), casually-aligned multi-projector desktop capable of running X11 applications.

I also developed high dynamic range (HDR) imaging code that used only a commodity camera and integrated it into Mersive's Sol Server platform which allowed them to perform better color correction in their multi-projector display products.

Graduate School
August 2003 - May 2007. Research Assistant at Center for Visualization and
Virtual Environments. University of Kentucky, Lexington, KY.
During my time as a research assistant, I worked in the field of computer vision and more specifically projector/camera systems. I published papers and attended several conferences and workshops. I was frequently consulted by students and faculty on hardware/software systems design and authored libraries which provided useful abstractions for cameras, projectors, and communication (both unicast and multicast with scatter-gather I/O) for the purpose of allowing less-experienced programmers to build vision and display systems easily and quickly.

Research Projects:
Telepresence: This project consisted of two systems each having a server with a camera, and four rendering clients with projectors comprising a tiled, geometrically calibrated, intensity-attenuated 4x1 display. The two systems were in different rooms and images from the camera in one system were sent to the server in the other system and multicast to the rendering clients who rendered their portions of the scene.

Terrascope: Designed and implemented a 9-camera system with both overlapping and disjoint views spread across an office suite for purposes of acquiring a ground-truth surveillance/activity dataset for the computer vision community.

Color Calibration/Object Recognition: Designed and implemented software to search the configurable color parameter space of an IIDC (firewire camera standard) camera to analyze the impact of different parameter configurations on object recognition.

High dynamic range (HDR) imaging: Developed tools for acquiring HDR images using a commodity camera at several different aperture settings, as well as tools for viewing slices of the images' dynamic range and performing image processing algorithms on them.

Color Correction in Projector Displays: This research was awarded an oral presentation at CVPR's Procams workshop in 2006. An extended version of it became my Master's thesis. I came up with a novel and general method for correcting color artifacts in projection displays that required no pre-measured environmental or equipment parameters and only 8 images of projected content. Unlike other common methods, my method does not unnecessarily augment regions of the display that are acceptable in order to fix aberrations in a distant part of the display. In this work, I also introduce a method for measuring the gamma of a display using a commodity camera.

Employment Before Graduate School
February 2002 - August 2003. Research Programmer/System Administrator at the Laboratory for Advanced Networking and later the Center for Visualization and Virtual Environments. University of Kentucky, Lexington, KY.
A jack-of-all trades position where I developed OpenGL and VRJuggler applications. I also researched and helped adapt the VRJuggler application system for use in the Metaverse immersive environment. I gained extensive experience with projector/camera systems throughout this job.

In addition to my programming duties, I was also responsible for ordering and managing heterogeneous computing infrastructure, configuring Cisco routers and switches, and using my broad knowledge of computing systems to advise students and faculty on the design of software/hardware systems, mostly using Linux and free and open-source software, for research projects.

2000-2001. Independent Consultant/Software Developer. Lexington, KY.
Wrote a variety of Perl/CGI/MySQL web applications for clients which included: user event tracking, automatic PDF report generation, automatic graph generation, a document search engine, etc. Also developed a Java-based storytelling/training application as well as a program to author modules for the training application.

August 1998 - June 1999. Developer at CommerceInc. Lexington, KY.
In charge of the software portion of the fulfillment end of the order pipeline. Wrote Perl/SQL applications that dynamically generated PostScript and groff code for all of our labels, paper media, emails, and fax media for customer interaction.

May 1997 - August 1998. Intern/Developer at Perot Systems Corporation. Lexington, KY.
Wrote Perl/CGI web applications for testing back-end services hosted on Solaris and AIX. Wrote log analyzers in Perl for various internal applications.

Skills/Tools/Software/etc

Image Processing/Vision/Graphics-specific:
OpenCV, OpenGL, National Instruments IMAQ/Labview camera API, ImageMagick, knowledge of image processing algorithms, IIDC/firewire camera standard, libdc1394 library for IIDC cameras, CGAL (Computational Geometry Algorithms Library), knowledge of mathematical optimization techniques

Other:
C/C++, Perl, Linux/Unix systems programming and internals, network programming, Matlab, Maple, Gnu Octave, Java, SQL, CGI, MySQL, web programming, Apache, Mac OS X, Gnu Scientific Library, multithreading/multicore programming, some experience with SIMD/MMX/SSE programming, some experience with Nvidia's CUDA, some familiarity with Python, Ruby, Objective-C

familiarity with non-mainstream languages: Common Lisp, Scheme, Smalltalk, ML, Prolog; interest in the academic field of programming languages and familiarity with programming constructs and features from high-level/experimental programming languages

Comfortable implementing algorithms/techniques from academic papers

References available upon request