

Editor: Daniel Zeng
University of Arizona and
Chinese Academy of Science
zeng@email.arizona.edu

GOLD: A Framework for Developing Intelligent- Vehicle Vision Applications

Massimo Bertozzi, Luca Bombini, Alberto Broggi, Pietro Cerri, Paolo Grisleri,
Paolo Medici, and Paolo Zani, *VisLab, Università di Parma*

To develop real-time vision applications for use in highly dynamic environments, such as automotive traffic, researchers must gather large amounts of data from different sensors and systems at different rates. Software

capable of real-time data acquisition, synchronization, logging, and—ultimately—data processing and visualization is fundamentally important to improving researcher efficiency.

The General Obstacle and Lane Detection framework supports different devices and makes it easy to add new system functionalities.¹ GOLD can easily become the engine for many automotive applications, and it could work in other application domains as well.

The GOLD framework

The use of vision in intelligent transportation systems has been in development for the past 20 years. It has evolved rapidly and is widely considered one of the most promising perception technologies.

Effective vision-based systems require developers to consider both hardware and software issues. Hardware issues include the computing engine and the sensing technologies—in vision systems, the cameras. Several camera technologies are available: daylight, far infrared (thermal), near IR, or even range cameras. The choice depends mainly on the specific application and cost-benefit analysis.

Software issues include the offline development and debugging software and the final system software. A rapid-development tool lets programmers focus on their application's specific vision problem without having to tend to other details. VisLab has developed GOLD to give programmers such a tool. GOLD includes APIs for common tasks such as I/O operations (acquisition from cameras and other sensors, sensor synchronization, network and file management, data graphical display, and so on) and low-level image-processing functions. A user-friendly GUI supports programming (see figure 1).

We originally conceived GOLD as the main engine to detect lanes and obstacles on board the Argo autonomous vehi-

cle.² After 15 years of continuous R&D, GOLD has evolved into a framework aimed at providing a complete set of tools for fast development of computer-vision applications. Moreover, GOLD runs the final onboard system. In spite of this complex functionality, GOLD's user-friendly GUI manages all the system applications. For example, you can visualize the data coming from different sensors, such as a stereo camera system or a laser scanner. Figure 2a shows laser-scanner data for the TerraMax prototype vehicle,³ one of only five vehicles to complete the 2005 Grand Challenge course and the only one to do so using a vision system for obstacle detection. Figures 2b and c show intermediate processing results for the laser-scanner data, and figure 2d shows the final processing results.

The GOLD architecture

Figure 3 shows the GOLD architecture and components. We conceived the architecture to ensure scalability and independence among the components. The whole system comprises several subsystem layers, which provide the different functionalities to developers. A cleanly defined interface makes it easy to add or remove subsystems, depending on the target project's specific needs. The framework views all applications as plug-ins that developers can easily remove or connect independently and port to different platforms.

The hardware abstraction layers mask the input device complexities from the programmer.

Developers can also easily produce highly interactive, consistent user interfaces through a set of widgets specifi-

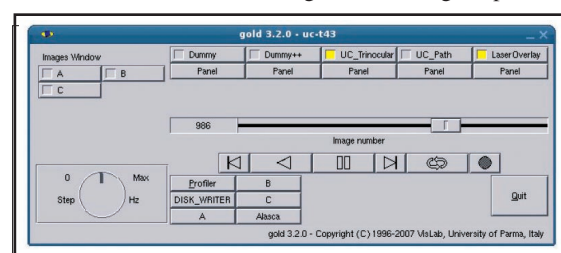


Figure 1. The GOLD GUI as implemented for the TerraMax prototype vehicle.



Figure 2. Examples of visual data outputs for the TerraMax prototype vehicle: (a) laser-scanner data, (b) top view of the area in front of the vehicle, showing lane markings and obstacles, (c) laser data mapped on the left camera image, and (d) the final results showing lane markings and obstacles after data processing and sensor fusion in real time.

cally designed to seamlessly integrate with the processing code. A windowing subsystem provides a simple, powerful set of drawing primitives that developers can use to display intermediate and final results during the debugging phases as well as the final output.

Building applications with GOLD

GOLD lets you build vision-based applications as independent plug-ins. Each application can take advantage of an evolving set of tools and debugging facilities.

The framework frees developers from the burden of performing low-level hardware management by offering an abstraction layer over real devices (that is, anything under the “Data acquisition API” subtree in figure 3). GOLD supports various classes of devices, such as different camera types (analog and digital), laser scanners, radars, vehicular controller-area-network data, and inertial systems. Applications can acquire data from multiple sensors with or without synchronization; this feature simplifies the development of data-fusion-oriented algorithms. Moreover, you can implement

virtual devices to access prerecorded data. GOLD also supports preprocessing of acquired data. Applications can perform common operations, such as image stabilization or distortion removal, and share the results with all the active plug-in applications.

A GUI layer provides graphic widgets and controls for interacting with applications, so developers need not concern themselves about available graphic back ends. A rich set of APIs lets them output final or partial results, producing graphics with very low resource consumption, thanks to a multithreaded client-server architecture.

Developers can use GOLD to record data sequences from multiple streams and play them back later for in-lab development. Applications can store the captured data in different formats, depending on the data’s nature and the application’s constraints. GOLD supports advanced playback capabilities, such as looping, stepping, jumping, moving forward or backwards at a given frame rate, and setting bookmarks. The team that developed GOLD applications at VisLab over the past 10 years also

engineered the playback capabilities with a view to guaranteeing flexibility in algorithm development and testing.

During playback, GOLD’s acquisition layer feeds the acquired data to the application through the same APIs used for acquiring data from real devices. Recorded data comes with an index file and a time stamp for each event. This information supports a time-based replay mode for handling problems involving time, such as tracking and synchronization. The playback can take into account the actual processing time and therefore emulate the system behavior in real working conditions.

Vision systems based on GOLD

GOLD has powered many systems developed at VisLab in the past decade. These include pedestrian detectors using visible and far and near IR images in projects sponsored by the US Army Tank-Automotive Command and by Volkswagen, Germany; traffic sign recognition for Marelli in Italy; vehicle detection for Fiat in Italy; and short-range obstacle detection for Volvo Technology in Sweden.

DARPA Urban Challenge

VisLab also used GOLD to develop artificial-vision systems for the 2007 DARPA Urban Challenge. In this project, we worked with Oshkosh Truck, Teledyne Scientific, Auburn University, and Ibeo.

The project’s vision software is spread over four high-performance dual-core PCs. It uses 11 cameras to sense the environment, providing a wide range of information: drivable path, lane position, long- and short-range front and rear obstacles, and overtaking vehicles and traffic at junctions.

The Urban Challenge scenario is complex, and the design efforts focus totally on reaching the levels of accuracy, robustness, speed, and completeness necessary to complete the mission. We weren’t addressing issues such as power consumption or hardware size. We selected PCs that were among the most powerful; each one hosts a full-featured copy of GOLD for both acquisition and real-time processing.

Monitoring boat speeds

In Venice, many watercraft move too fast through the channels, generating waves that erode the seacoast. Preserving this coast is

important to saving the entire ecosystem that characterizes Venice.

For this problem, VisLab developed a system that determined boat speeds and displayed them to deter excessive speeds. The system uses a high-resolution camera on top of a 10 m pole to monitor an area up to 200 m distant. The camera is connected to an off-the-shelf PC together with a large LED panel that shows the boat speed. The system stores all the data (speed, date, and time) of the identified watercraft in a log file and can act as a Web server, providing information in real time.

This application requires only part of the GOLD system: low-level vision filters, the hardware-abstraction layer, and real-time data management. Ultimately, we plan to port the whole processing system on a smart camera powered by an ARM processor and a Linux-embedded operating system.

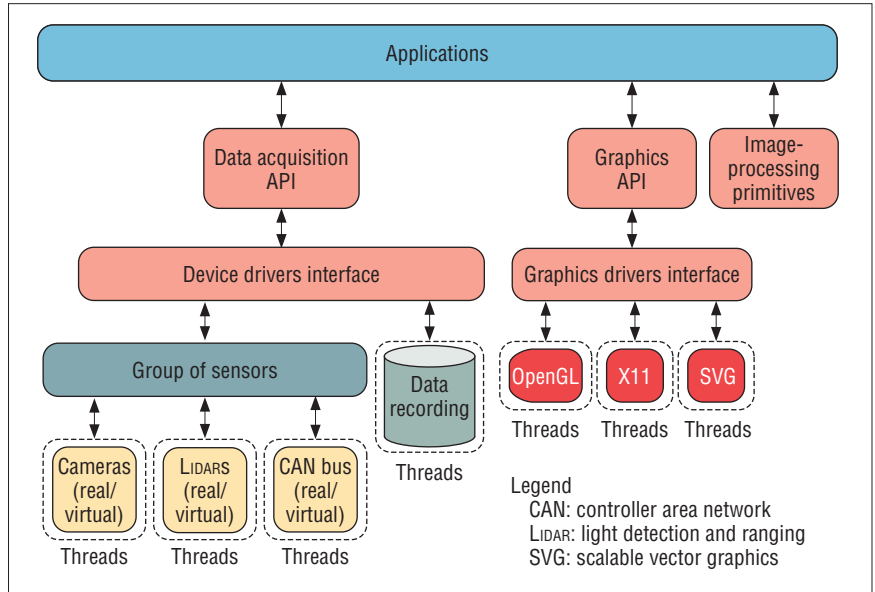
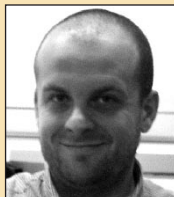


Figure 3. The GOLD architecture. The different colors represent software blocks belonging to same logical level.

GOLD's support for rapid application development and debugging as well as fielding complex vehicular systems is unique. There are some similar projects. For example, MOOS provides simple, open-source, cross-platform software for mobile robotics research (www.robots.ox.ac.uk/~pnewman/TheMOOS) and MIDAS 4.0 is a commercial video- and data-capture product (www.xcitex.com/html/midas_description.php). However, these systems don't have GOLD's flexibility, and they're usually devoted to only one product-development phase. After 15 years of software development, GOLD represents a fundamental instrument for vision algorithm development, and it's the key to VisLab success in the international research arena. ■

References

1. M. Bertozzi and A. Broggi, "GOLD: A Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection," *IEEE Trans. Image Processing*, vol. 7, no. 1, 1998, pp. 62–81.
2. A. Broggi et al., *Automatic Vehicle Guidance: The Experience of the ARGO Autonomous Vehicle*, World Scientific, 1999.
3. A. Broggi et al., "The Single Frame Stereo Vision System for Reliable Obstacle Detection Used during the 2005 DARPA Grand Challenge on TerraMax," *Proc. IEEE Int'l Conf. Intelligent Transportation Systems (ITSC 06)* IEEE Press, 2006, pp. 745–752.



Massimo Bertozzi is an assistant professor in the Università di Parma's Dipartimento di Ingegneria dell'Informazione and a researcher on the Argo project. Contact him at bertozzi@ce.unipr.it.



Pietro Cerri is a temporary researcher at Vislab in the Università di Parma's Dipartimento di Ingegneria dell'Informazione. Contact him at cerri@ce.unipr.it.



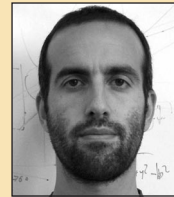
Luca Bombini is a PhD student in information technologies at the Università di Parma and a temporary researcher in the Dipartimento di Ingegneria dell'Informazione's Artificial Vision and Intelligent System Laboratory (Vislab). Contact him at bombini@ce.unipr.it.



Paolo Grisleri is a temporary researcher at Vislab in the Università di Parma's Dipartimento di Ingegneria dell'Informazione. Contact him at grisleri@ce.unipr.it.



Alberto Broggi is a professor in the Università di Parma's Dipartimento di Ingegneria dell'Informazione and the founder of VisLab. Contact him at broggi@ce.unipr.it.



Paolo Medici is a temporary researcher at Vislab in the Università di Parma's Dipartimento di Ingegneria dell'Informazione. Contact him at medici@ce.unipr.it.



Paolo Zani is a PhD student in information technologies at the Università di Parma and a member of the Dipartimento di Ingegneria dell'Informazione's Vislab. Contact him at zani@ce.unipr.it.