VizlerControl: Eine Modulbasierte Umgebung zur Steuerung von Geräten auf der Basis von Bildanalyse.

VizlerControl: A Module-based Environment for Device Control based on Image Analysis

Horst Eidenberger
Vienna University of Technology, Institute of Software Technology and Interactive Systems,
Favoritenstrasse 9-11 – 188/2, A-1040 Vienna, Austria
Tel.: +43-1-58801-18853, Fax: +43-1-58801-18898
E-mail: eidenberger@ims.tuwien.ac.at, Internet: http://www.ims.tuwien.ac.at/~hme

Abstract:

The VizlerControl project aims at the development of a standardized, module-based environment for hardware device control. The term Vizler stands for video analyzer. The name points at the underlying intention of the project: controlling devices based on the information provided by *visual capture devices* (video and photo cameras). The VizlerControl system consists of a pipeline of modules. Figure 1 depicts the modules and the dataflow.

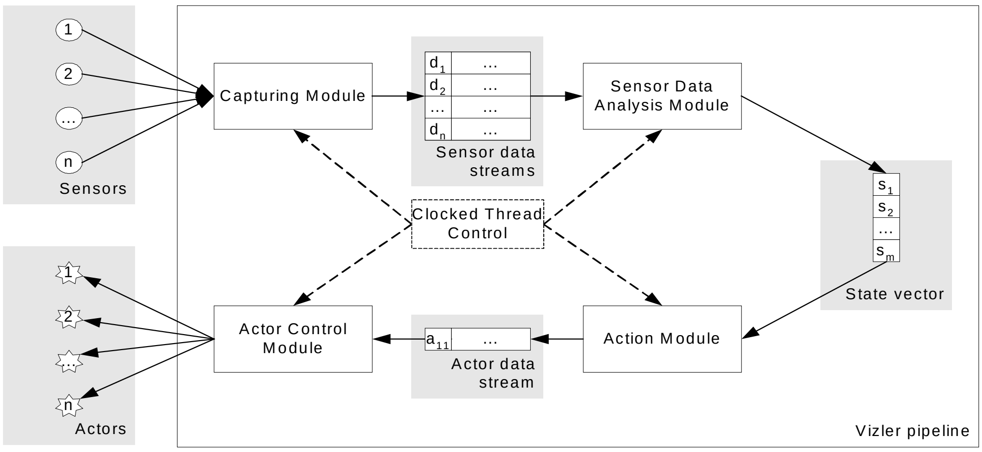

Figure 1. Vizler processing pipeline.

Input is provided by an arbitrary number of sensors. Sensors are used to detect relevant objects in the environment. Sensors can be visual devices as well as magnetic sensors, temperature sensors, infrared sensors and many others (see [1]). In VizlerControl we are mainly working with two types of sensors: video cameras and photo sensors. The Capturing Module (CM) captures the signals from all sensors and creates a real-time data stream for each sensor. Data streaming in VizlerControl is based on the Real-time Transfer Protocol (RTP, see [3]).

The Sensor Data Analysis Module (SDAM) analyzes sensor data in real-time and derives state values s_m for m relevant objects in the environment. Based on these state values, the Action Module (AM) derives control commands that are sent in a stream to the Actor Control Module (ACM). The ACM provides a standardized interface to the various actors. Actors are used to manipulate relevant objects in the environment. In VizlerControl we use motors and graphical user interfaces as actors. Motors are controlled by programmable interface devices. Communication is based on the serial RS232 interface. All modules are clocked and synchronized by the Clocked Thread Control.

Each module consists of an extendible set of software objects (classes and interfaces). The software objects can be stitched to a pipeline using a well-formed XML-document ([5]). As part of

VizlerControl, we have developed a Document Type Definition (DTD) with the name VizlerControl Markup Language (VCML). VCML standardizes the definition of sensors and actors, the configuration of pipelines, the input and output of software objects and the event-based time control in the Clocked Thread Control. Timing and synchronization elements of VCML were partially taken from the W3C SMIL specification (see [4]).

The focus of the paper is on the implementation of the SDAM for a specific application. In this application, a tabletop soccer device is partially controlled with VizlerControl. A digital video camera is used as the only sensor. Motors are used to control the players. With this prototype it is possible for a single player to play tabletop soccer against a computer. We chose tabletop soccer, because it is a very fast game (non-deterministic very fast ball movement, etc.) on a limited, small area.

Figure 2 shows the camera view. In this example we use a 320x240 pixel input video stream. This input stream is taken from a consumer MiniDV digital video camera. The PAL DV-compressed video stream is converted to uncompressed 24bit raw RGB-video by the CM. The SDAM is used to find the position of the ball in each input video frame. The state vector has three elements: X- and Y-coordinate of the ball position and the likelihood of correct detection. Ball tracking is based on fast color segmentation. In our experiments, yellow has turned out to be the optimum ball color. The likelihood of correct position detection is derived from the number of detected contiguous pixel with matching color. Ball detection is limited to a square region of interest. In Figure 2 the black trapezoid visualizes the border of the ball search region of interest and the point and the black circle around it show the detected the likely ball position for this frame. This method of ball detection has turned out to be fast and effective.

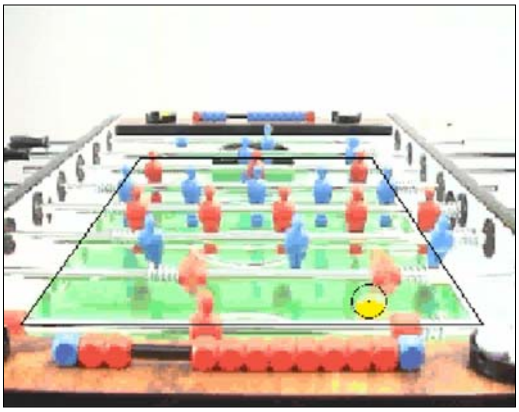

Figure 2. Vizler video tracking.

VizlerControl is implemented in Java as an open class framework. Modules are represented as interfaces. Software objects implement the methods and resources of these interfaces as classes. The SDAM for video cameras and the RTP streaming components are based on the Java Media Framework (see [2]). We chose Java, because it supports state-of-the-art software development processes (like the Rational Unified Process, etc.) and offers powerful image and video processing capabilities (Java2D, Java Advanced Imaging, etc.). Additionally, the performance of Java is not as bad as often asserted.

The tabletop soccer prototype was implemented with support of A.u.S. Spielsysteme GmbH, Vienna, Austria (http://www.aus.at/).

References

[1] Altenburg, J., Altenburg, U. Mobile Robots (in German), 2edn., Hanser Verlag, München, Germany, 2000.
[2] Java Media Framework Website. http://java.sun.com/products/java-media/jmf/index.html.
[3] Real-time Transfer Protocol. http://www.ietf.org/rfc/rfc1889.txt
[4] World Wide Web Consortium: SMIL 2.0 Specification. http://www.w3.org/TR/smil20/
[5] World Wide Web Consortium: XML Website. http://www.w3.org/XML/