Image processing workflow middleware to archive high performance and usability

K. Iwata¹, Y. Satoh¹ and I. Kojima¹
¹ National Institute of Advanced Industrial Science and Technology (AIST)
AIST Tsukuba Central 2, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568 Japan

Abstract - We develop Lavatube as an image processing workflow middleware for efficient research and development. Lavatube is an object-oriented framework optimized for constructing a computer vision system, particularly a video and image processing system. Lavatube enables a description of a processing extension by combining various functional components. Since the data flow is easy to describe by graph-connecting icons on a GUI, a system can be created intuitively. For efficient and clearcut system construction, Lavatube provides functions for the dynamic generation of parameter setting dialog boxes and for perpetuation by XML. Lavatube can also be used cloud computing to process large amounts of data. Some actual image processing cases as examples are also introduced.

Keywords: Image Processing, Middleware, Workflow, User Interface

1 Introduction

A computer vision system captures information from the external world as image data by using a camera and other devices, and analyzes the images on a computer for studying or measuring [1]. This kind of system is used as a major means of inspection, especially for semiconductors and electronic boards, because objects can be measured without contact. The range of applications, such as security, robot vision, medicine, welfare, and sports, has been growing even more in recent years.

A computer vision system that is expected to be applied so widely requires high-level knowledge and programming techniques for its design and adjustment. So, at a company or a university, what kind of abilities should be acquired to learn this system from the beginning? There is a lot of substantially important knowledge, such as statistics and geometry. In reality, however, he or she faces such problems as difficult programming for image acquisition from a camera or time-consuming analysis of a predecessor's program that requires great overhead.

Even an experienced person tends not to be committed to programming because it requires a lot of time to construct a framework where arbitrary processing can be visualized and various parameters can be adjusted in real time. This consequently makes it necessary, yet sometimes impossible, to evaluate installed algorithms and set parameters satisfactorily.

To solve these problems, we developed Lavatube as a image processing middleware. Lavatube has the following advantages:

- A system can be constructed easily by graph connecting icons that express various functions on a graphical user interface (GUI), such as the one shown in figure 1.

- By arranging icons with the capturing function, a USB camera and various video files (AVI, MPEG, etc.) can be used as input. This realizes online and offline experiments using previously recorded videos on a single platform.

- Parameter setting dialog boxes can be generated dynamically and adjusted by viewing of the processing results online.

- Persistence by XML enables storage and reproduction of working environment.

- The framework operates at high speed with extremely small overhead. Since parallel flows are automaticallythreaded for parallel execution, this programming makes processing even faster than ordinary programming.

Fig.1 A GUI of Lavatube
Previously, the image processing environments, such as Khoros [2] and XITE [3] have been developed. More recently however, many visual programming environments for processing flows visually through this type of GUI have been available commercially. For example, MAX/MSP [4], which is mainly used by artists, is a visual programming environment extended from voice processing to video processing. Other marketed products include MATLAB/Simulink [5] for simulation and AVS/Express [6] for visualization. These are basically visual programming environments in which existing modules are combined to create a processing flow. If there are not enough modules, extension modules are created by using C or other languages. Since existing modules are not adequate for creating new algorithms, programmers need to create extension modules.

On the other hand, workflow systems have been developed to manage a business process [7] or scientific studies [8]. The workflow systems can be performed with the description of the cooperation of a variety of functions and services. However, these systems are not designed to be easy to use in image processing.

Lavatube was developed not as a visual programming environment via a GUI but rather as an object-oriented framework optimized for video and image processing. Lavatube is designed so that users can develop original function extensions easily by combining various functional components. These patches are compatible with GUIs, XML, and other support functions, and realize an efficient research and development environment, in which such functions are immediately available as a user interface, visualization, and perpetuation.

Over the past few years, large amounts of data are being accumulated. Recently, cloud computing has been attracting attention as a useful infrastructure for the analysis of such data. We have developed a middleware to build a workflow of image processing on the cloud.

This paper outlines the Lavatube functions and introduces actual cases.

2 Outline

Lavatube is an object-oriented framework that supports research on video and image processing, computer vision, and also trial application programming and tuning. The characteristic functions of Lavatube are described below.

2.1 Description of Data Flow by the GUI

In the GUI of Lavatube, each process (called a "work patch") is expressed as an icon. By connecting the input and output of each icon using a mouse, a data flow can be described very easily. Figure 2 shows an example of a data flow description. In this example, the image output from the USB camera is connected to the input of the contour detection filter.

This type of GUI makes it easy to partially modify and add processes. For example, online experiments and offline experiments can be easily realized in the same environment by switching the source of image input immediately to a camera or a video file, as shown in figure 3.

2.2 Extensibility

A program is described in small units called patches, and Lavatube operates by interpreting these patches automatically. As figure 4 shows, work-patches are created by combining patches of functions such as data I/O, parameters, and by describing a processing procedure on the object interface. Any user experienced in C++ programming can easily create work-patches of arbitrary functions.

At present, Lavatube has functions for image capturing from a USB camera, basic filtering, and arithmetic operations. The
high extensibility allows for easy addition of functions to use other types of cameras.

2.3 Storage and Reproduction of Working Environment by XML

Lavatube can output a working environment including a data flow and parameters to an XML file for storage. The working environment can be completely re-produced by loading the XML files. As the system becomes complicated in ordinary programming, this processing becomes time-consuming and often causes a programming error. In Lavatube, each patch has a function that can be realized easily by separating the description. This function enables verification by reproducing the environment, as well as later additions or modifications of functions.

2.4 Parameter Setup via the GUI

Lavatube provides a GUI through which parameters can be adjusted easily. Since dialog boxes, such as figure 5(a), are created dynamically, programmers are free to use GUIs for additions or modifications without using a GUI builder or other software. The dialog boxes are automatically generated by describing any parameter in parameter function patches. Since parameter sets determined via the GUI are stored in an XML file, such as figure 5(b), a description of constant parameters can be separated from the source code to improve program maintainability.

2.5 Visualization of Operation Status

The real-time display of image data on a data flow allows the user to visually check the operation status sequentially for efficient parameter tuning and other tasks.

For real-time demonstration, Lavatube is designed to optimize the processing overhead and to operate the constructed system extremely quickly. The automatic parallel processing of tasks in each work patch is optimum for multi-core processors, which are becoming more widespread. Figure 6 shows a parallel operation example. In this figure, 25 tasks are operating on 8 cores PC.

3 Cloud-based system

The first version of Lavatube was an application for Windows. To take advantage of cloud computing, we developed Lavatube 2, which has a WEB-based user interface and cloud-based execution engine. The developed software, Lavatube 2, is divided into two parts: a user interface, Skylight, and an execution engine, Deepcave. The execution engine is located on the cloud and the user interface is provided from the Web browser. Figure 7 shows the configuration of Lavatube 2. This configuration has the following benefits. First, installation on a PC is not required, allowing the user to use Lavatube 2 independently of the equipment environment, including the performance of the PC and the operating system. Skylight is based on the latest HTML5 Web technology and allows the user to easily develop, with simple mouse operations in the Web browser window, a complex image analysis system consisting of programs to perform various processes and procedures. Data analysis by the execution engine Deepcave is performed by the cloud but not by the user’s browser. This makes it possible to process massive amounts of data at high speed, making use of the computing capability of cloud computing.
4 Examples

4.1 Wrapping external libraries

By wrapping OpenCV and other external libraries as work patches, Lavatube can handle them easily. Figure 8 shows an example of detecting a human face by using the Viola-Jones face detector [9]. The left-side image is a work patch captured from a camera. According to the flow from there to the lower stage, color-to-gray image conversion and facial detection patches are connected. Since the facial detection patch outputs the position co-ordinates and dimensions of a face, it is connected to a work patch that draws a circle at the face position. This work patch is created to receive image and coordinate inputs.

Figure 9 shows an example of optical flow estimation by using the Lucas-Kanade method. Because optical flow estimation requires current image and previous image, a work-patch that buffers previous images is connected to an optical flow patch.

4.2 Satellite image processing

The user can perform a range of image processing tasks on obtained image data by using the built-in work patches in Lavatube 2. Figure 10 is an example of a change detection system of satellite images. Work-patches to support various services based on OGC [10] and access to geospatial information services. Figure 11 shows a schematic of the cooperation of geographic information processing be-tween Lavatube and OGC W*S services. OGC CS-W is a retrieval system for contents or services. URLs of OGC WMS services. A WMS work patch request the map data of the obtained URL and the bounding box from the WMS service.

OGC CS-W is a retrieval system for contents or services. Figure 12 shows the retrieval of ASTER [11] data archive on GEO Grid [12]. A bounding box is specified using a Google Earth plug-in. A search query including the bounding box, dates and cloud covers is posted to a CS-W server, which lists the responses and presents them to the user.

Left of figure 13 shows data from June 2012 and June 2011 in the coastal area of northeast Japan. Right of figure 13 shows the result of change detection. The figure shows signs of change toward recovery one year after the Great East Japan Earthquake of 2011. The construction of this kind of change detection systems is not easy, because the satellite data exhibit variability in quality due to differences in vegetation and weather conditions. Thus, a change detection system requires trial and error experimentation using a combination of methods and parameters. Users can develop systems by trial and error using Lavatube 2.
5 Conclusion

This paper outlines Lavatube as a workflow middleware to support trial programming and research for computer vision or image processing by a GUI and XML, and introduced some cases as examples. We are working on middleware that will provide a new infrastructure for geospatial information research.

Acknowledgment

This work was supported in part by MEXT/JSPS KAKENHI 24700082.

References


