SOFTWARE SYSTEM FOR DEVELOPMENT OF ALGORITHMS FOR VIDEO PROCESSOR

E. Kalabukhov, M. Tatur
Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus
E-mail: tatur@i-proc.com; kalabukhov@bsuir.by

1. Task of creating the system

Video processing system can be built using both universal and specialized calculators – video processors. The hardware part of the specialized system typically includes a video camera, a video processor and monitor (or video server).

In addition to creating the actual hardware of the video processor, you typically need to create firmware that translates the video stream received from a camera to achieve a certain effect (for example, increase the contrast, sharpness, performing filtering etc.).

Perform creation and debugging of firmware directly to the video processor is quite a long process (creation – flash – testing – analysis – modernization, etc. the cycle). It is time-consuming and requires, in some cases stable video input.

Besides creating firmware for a number of video processors depends on the hardware platform, (for example, resources, feasibility of processing operations), although the processing algorithm remains almost total.

Our approach is to use a software system for creating and debugging the initial idea of the algorithm. At the same time, this step does not use a dedicated video processor and runs on a personal computer, although the formation of the algorithms should be carried out taking into account the special requirements for data format and permissible operations with them. In such a software system source and the received data is stored in the video files (as .avi video containers). All this allows simplifying the development and testing of algorithms, allows accumulating base general algorithms, which can then be quickly adapted to the desired structure of the video processor, makes it easier to compare the results and the complexity of implementing various versions of the algorithms.

2. System structure

The software system consists of the following main components (see figure 1):

- main module, which is a user interface and organizing work with the input and output video, and user settings for processing;
- video frame processing modules; each module (the effect) is a dll with a fixed interface, but rather in terms of the implementation of the free video frame processing;
- communication component (special DirectShow filter in format .ax, see Figure 1, component 3) for exchange data between the main module and the video frame processing module to support the processing of video sequences on DirectShow technology.

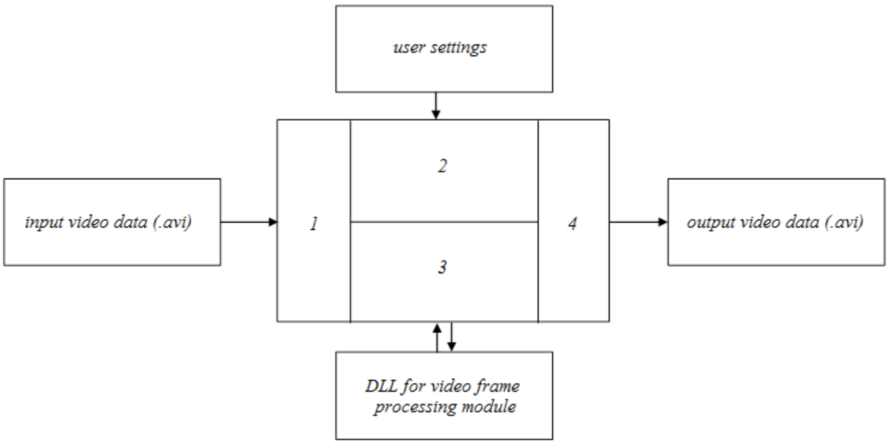


Figure 1. The software system (1 – video input interface; 2 – user interface (GUI); 3 – interface for video frame processing modules; 4 – video output interface)

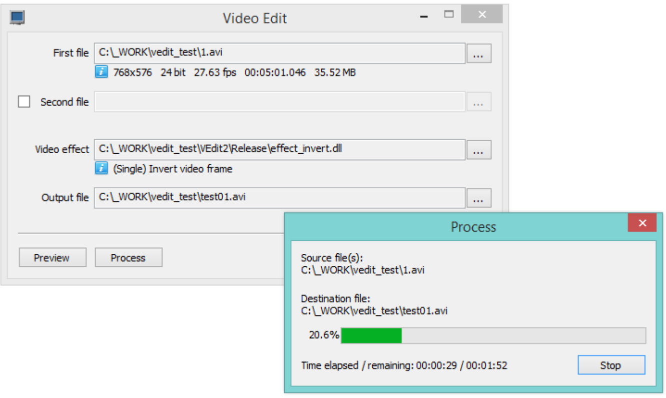
3. Features of the implementation system

The system is implemented in C++ (created use MS Visual Studio 2008) for Windows operating systems (tested on Windows XP, 7, 8).

The basis of the video processing technology incorporated use DirectShow, with the main program module of the system (see figure 2) is easy to use. Since the system is intended for complete processing of the input video data, the specialized tools to select fragments of video data therein are not provided. System settings allow you to select the codec to output video.

Effects for the treatment selected from the set pre-existing dll for video frame processing. The results of one of the test effect are shown in figure 3. Currently supported video processing only one filter per pass, because it allowed a few to simplify the implementation of the system, although it reduced the overall functionality.

To implement the new processing modules you need to create new dll with strictly defined structure that receives program access to the image frame. It is also one of the features of the system – programmer is separated from to work directly with the video and with effects modules registering in the system, which greatly simplifies the creation and debugging of effects.



Video Edit

First file C:_WORK\vedit_test\1.avi

768x576 24 bit 27.63 fps 00:05:01.046 35.52 MB

☐ Second file

Video effect C:_WORK\vedit_test\VEdit2\Release\effect_invert.dll

(Single) Invert video frame

Output file C:_WORK\vedit_test\test01.avi

Process

Source file(s):
C:_WORK\vedit_test\1.avi

Destination file:
C:_WORK\vedit_test\test01.avi

20.6%

Time elapsed / remaining: 00:00:29 / 00:01:52

Figure 2. The user interface of a software system.

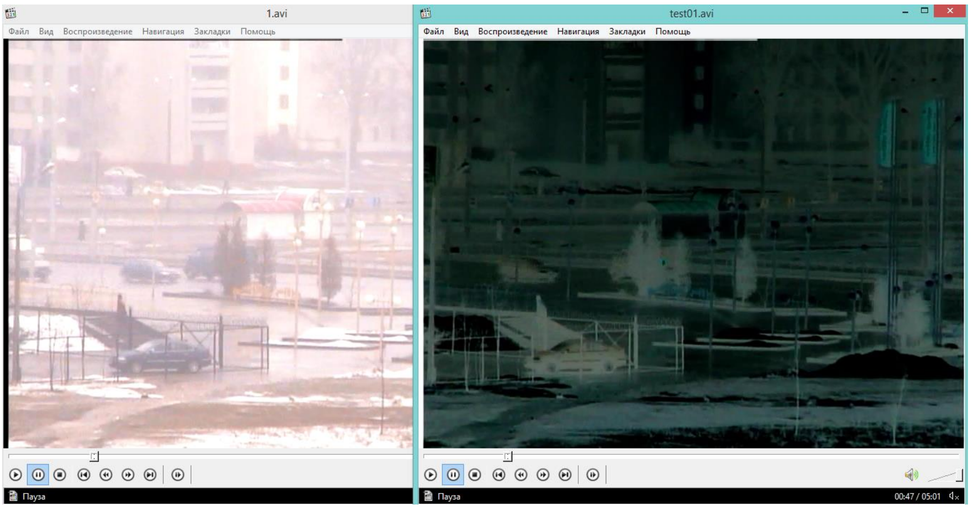

1.avi
Файл Вид Воспроизведение Навигация Закладки Помощь
Пауза
test01.avi
Файл Вид Воспроизведение Навигация Закладки Помощь
Пауза
00:47 / 05:01

Figure 3. Example of the effect of inversion

4. Using the system

The resulting program system allows:

- get open to analyze and debug algorithms for video processing, because creation of algorithms is quite low level programming, which contributes to further more rapid and adequate translation of the algorithm into the firmware of the video processor;
- implement fairly sophisticated video processing algorithms, because restrictions on treatment exist only in the form of an agreement on the format of the data and a set of operations that must use the programmer, which created algorithms;
- store received algorithms in libraries, but the registration of the modules handling personnel in the operating system is not made that quite effectively when effect on debugging;
- to analyze the result of the visual comparison of algorithms results;
- create a video processing results, which makes it easy to demonstrate the progress and development of algorithms and compare results between different versions of the same algorithm or its settings;
- process as individual frames or as stereo sequences from different video streams, that the latest version can be applied for comparative analysis.

The resulting system at this stage is mainly focused on pre-processing of video data, but may be supplemented by other technologies of video analysis, because it depends only on the operation of the module of video frame processing.

The system is currently used to create algorithms for nonlinear filtering of the video, which will be further implemented in the firmware of the original video processor, built on FPGA Xilinx.

In the future we plan the creation sdk to simplify programmer work with an image frame for effect module, create a library of specialized modules for generating statistics on processed video data, and update of the system for the use of several processing modules in the sequential processing a frames of video data.