DISPLAY INSPECTION SYSTEM

Tomáš Babinec, Pavel Čík

Doctoral Degree Programme (2), FEEC BUT

E-mail: xcippa00@stud.feec.vutbr.cz, xbabin01@stud.feec.vutbr.cz

Supervised by: Ilona Kalová

E-mail: kalova@feec.vutbr.cz

Abstract: This paper deals with the design of automated inspection system for graphical environments in display devices. Visual human machine interfaces play a very important role in our everyday live. Their development usually includes design of complex graphical objects and menu structures. Consistency check and functionality inspection is an important part of such a development. Successful automation of the inspection process does significantly reduce the time required for the development, making it cheaper and more dynamical.

Keywords: computer vision, image processing, display, screen, inspection, GUI, consistency

1. INTRODUCTION

For humans visual perception is the most important source of information. Therefore one of the fundamental characteristics of a human machine interface is the ability to communicate in graphical form. Images have the ability to encode large amounts of data in a very efficient and easy to understand way. Nowadays various kinds of screens (fixed segment, dot matrix, LCD, TFT, OLED, E-INK ...) can be found in all sorts of electronic devices (cell phones, PDAs, calculators, touch panels ...).

The mass production of display panels has triggered the need for screen quality inspection systems. Functionality of most of the proposed solutions (for example [3]) is based on image processing and machine vision algorithms and concentrates its effort on detection of faulty hardware features.

Completely different category of inspection mechanism is required by end-user device manufacturers and graphical user interface (GUI) developers. Their never ending competition for better looking, more user friendly and intelligent software environments leads to extremely complex screen content. The content is usually compiled from sophisticated menu structures and other graphical objects, which change dynamically according to actual situation and user input.

Such GUIs require rather extensive check of their consistence and functionality. The test procedures are monotonous and time consuming. As long as they are performed by human operators, the test results can be affected by operator's fatigue or other influences that are difficult to evaluate. Therefore there are attempts to minimize or completely replace human test interaction by automated inspection systems.

This article deals with the above presented problem and describes the development of semiautomatic inspection system for GUIs displayed on various screen devices.

2. SYSTEM DESIGN

The designed solution combines hardware (HW) components and software (SW) modules into complete inspection system. The development is focused at imitating humanlike behaviour during test process. Therefore the system facilitates visual evaluation together with the possibility of online human input simulation as a reaction to currently displayed information.

2.1. REQUIREMENTS

» Maximum separation between inspection system and tested devices. This approach minimizes mutual influence and error propagation.

» A simple and rapid way to script the test procedures, provided by well organized object-oriented scripting interface.

» Uncomplicated adjustment and functionality expansion.

» Minimum need for human interaction.

2.2. HARDWARE SET-UP

The most essential HW element of the proposed inspection system (viz. Figure 1) is an Imaging Source industrial camera connected to a PC. A 41BU02 camera model has been chosen. It has a 1280x960 RGB CCD sensor with Bayer encoding capable of 15 frames per second. This is sufficient for the inspection of majority of the various embedded systems' displays.

In order to properly position the camera (according to the dimensions of the tested display and available optics) an adjustable stand has been used. In case of inspecting passive screen devices (e-ink...) additional light source may be required.

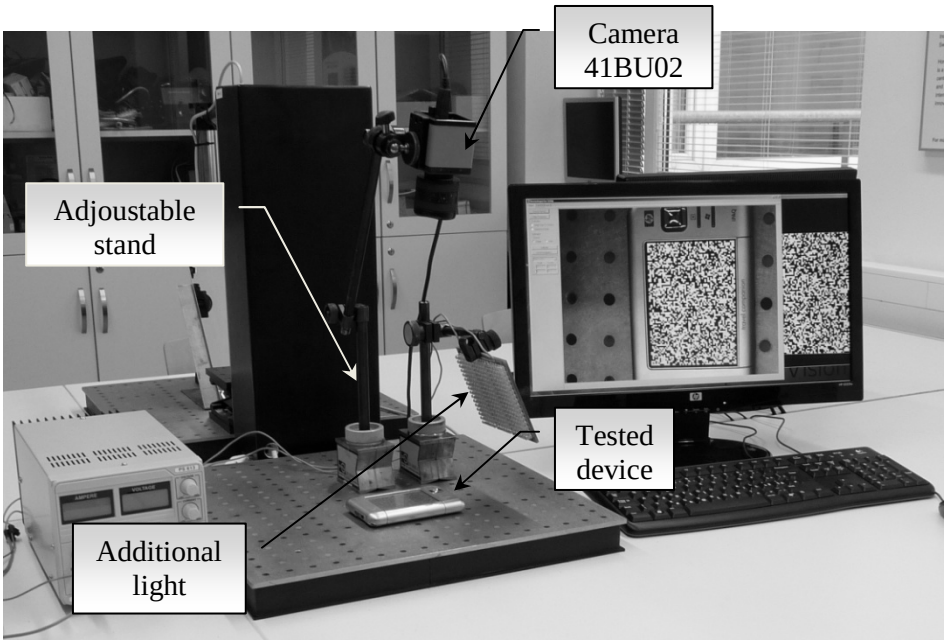

Figure 1: Inspection system HW setup

2.3. SOFTWARE ENVIRONMENT

In order to provide all of the expected functionality, the inspection SW environment does include modules depicted in Figure 2. There is a camera communication module entitled "UsbCam" implementing camera properties management and image grabbing. "Screen Inspector SetUp" is a semiautomatic module which allows effortless system adjustment and calibration. To enable online and automatic control of the inspected device a custom built functionality represented by "Input Simulator" module may be linked to the system.

The actual image processing and test scripting happens inside the three-level module. It is composed of LowLevel library with high-performance image processing algorithms written in C/C++ language. HighLevel library was programmed using CLR C++ and its purpose is to wrap C/C++

algorithms to be used in managed code under .NET environment. This intermediate level also provides an object oriented interface for C# test procedure scripting.

Since the final test procedure coding is supposed to be straight forward and rapid, C# object oriented approach appears to be a satisfying choice.

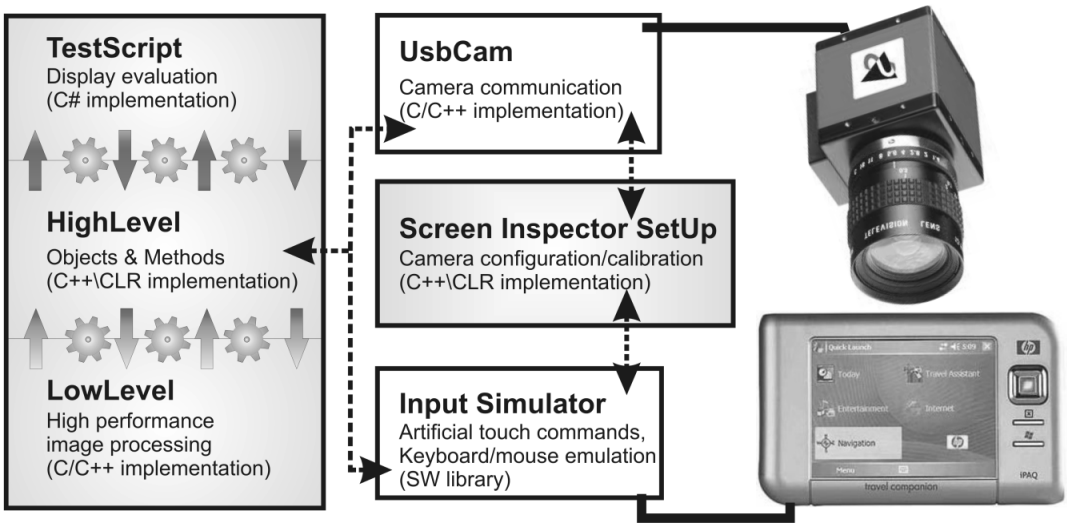

Figure 2: Inspection SW architecture

2.4. DISPLAY DESCRIPTION

To describe required screen content, its structure and hierarchy a simple but rather flexible XML scheme has been designed. It is based on an idea that for every basic screen element, which should be inspected, there has to be either some defining bitmap or a rectangle area with character string.

Combination of these fundamental features and information about screen element's membership in some defined parental object (panel, menu, screen…) creates a coherent device GUI description. This approach also enables simple functionality expansion.

Since display description is a rather specific and device dependant matter, the inspection SW library provides only elementary methods, which are supposed to be used in order to derive more complex functionality for detection and inspection of specific screen objects. Thanks to this quality the end-user is able to independently boost the provided inspection library with no or minimal support from the inspection system developers.

3. OPERATION & RESULTS

The designed inspection system has two modes of operation. These are calibration mode and inspection mode. First the calibration and HW set-up has to take place (in more detail described in section 3.1) in order to adapt the system to actual conditions. Afterwards the system is ready for test procedure execution (inspection mode - in more detail section 3.2).

3.1. SYSTEM CALIBRATION

This is a semi automated procedure. The required operator's are as follows:

- » Hardware set-up
  - ➢ Adjustment of mechanical properties
    - ▪ Inspected device positioning; camera – display measuring distance set up.

- Adjustment of optical properties and lightening conditions
  - Mechanical shutter set-up and focusing, surrounding lightening adjustment

» Software set-up

- Image properties
  - Shades of gray/color imaging; frames per second rate; gain, white balance and gamma correction adjustment.
- Camera calibration
  - Automatic screen location and projective transformation identification.

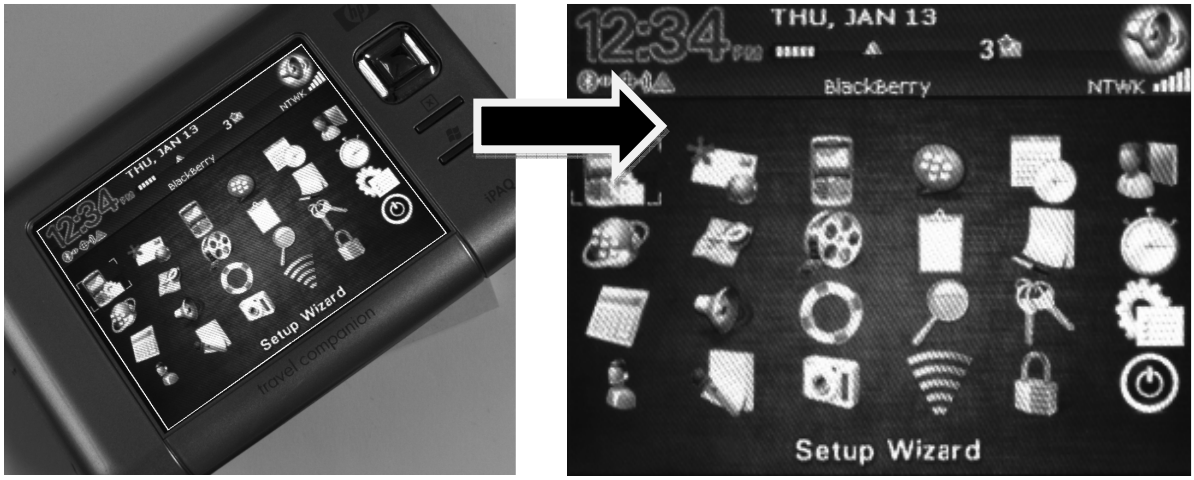

Figure 3: Camera calibration example (Left: original image; Right: located screen area after rectification).

In order to simplify the subsequent display inspection the system calibration includes a screen plane to camera frame projection matrix identification [1]. Result of this operation is shown in Figure 3. This calibration step can significantly reduce inspected area of the camera frame and also ensures a normalized size of the images which are passed to further processing.

$$\begin{bmatrix} mX_D \\ mY_D \\ m \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} X_S \\ Y_S \\ 1 \end{bmatrix} \quad (1)$$

Equation (1) introduces homography mapping between 2D source point $[X_S, Y_S, 1]$ (immediate screen point) and 2D destination point $[mX_D, mY_D, m]$ (point in the camera frame imaging the screen). The unknown elements $h_{11} - h_{32}$ describe the projection transformation with 8 degrees of freedom. The transformation (1) utilizes homogeneous coordinates' notation which allows representing planar rotation, translation and perspectivity by simple 3x3 matrix multiplication [1].

For successful calibration at least 4 screen-to-camera corresponding points are to be required. However to minimize possible image uncertainties much denser mesh of points like a binary white noise pattern would be appropriate. Further information on coordinates mapping and homogenous coordinates' information can be found for example in [1, 4, 5].

3.2. DISPLAY INSPECTION

Results of basic inspection functionalities are displayed in Figure 4. To compare whole screen areas a cosine angle local similarity criterion [2] has been used. Feature localization was implemented

using patter matching methods and there is also the OCR functionality available among the basic system library algorithms.

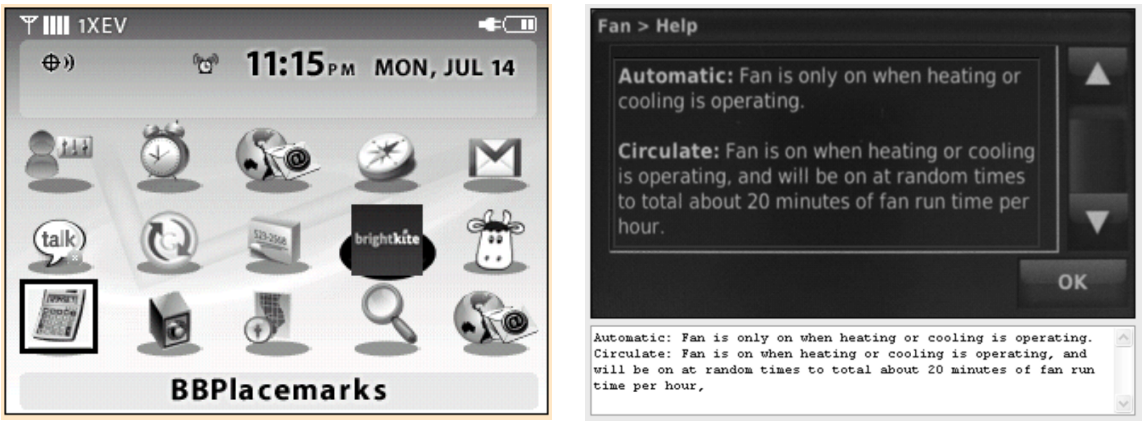

Figure 4: Inspection mode (Left: object location; Right: OCR algorithm)

4. CONCLUSION AND FUTURE WORK

The introduced inspection system represents a working concept for machine vision based automatic GUI functionality and consistency check. It fulfils all the initial requirements stated in section 2.1 of this paper. The image processing is based on OpenCV library algorithms, which ensures effective and fast execution. Test procedure scripting is intuitive and rapid thanks to the C# scripting interface.

In order to improve reliability and extensibility of the created system, the utilized image processing algorithms will be revised and a superior scheme for screen GUI description is going to be proposed.

ACKNOWLEDGEMENT

This work was supported by grant „Modern Methods and Approaches in Automation“ from the Internal Grant Agency of Brno University of Technology (grant No. FEKT-S-10-12).

REFERENCES

[1] Penna, P. & Patterson, R. (1986). Projective geometry and its applications to computer graphics, Prentice-Hall, ISBN 0-13-730649-0, USA

[2] Jan J. *Medical Image Processing Reconstruction and Restoration: Concepts and Methods.* Boca Raton: Francis & Taylor 2006, ISBN 0-8247-5849-8

[3] Black, S.E.; Goodman, R.L.; Wood, K.N.; , "Multi-Channel Deep-Memory Digitizing Architecture for Automated Inspection of Large Composite Surfaces," Autotestcon, 2006 IEEE , vol., no., pp.558-564, 18-21 Sept. 2006

URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4062440&isnumber=4062301X ML tutorial

[4] Bradski G., Kaehler A. *Learning OpenCV*. Sebastopol: O'Reilly Media, Inc. 2008, ISBN: 978-0-596-51613-0

[5] Žára J., Beneš B., Sochor J., Felkel P. *Moderní počítačová grafika*. Praha: Computer Press, 1998. ISBN 80-251-0454-0