Developing visual servoing applications over the Windows7 mobile platform

Marco Perez-Cisneros, Laura Lopez-Lopez, Erik Cuevas, Daniel Zaldivar, Marco Cedano-Olvera

[1] División de Electrónica y Computación, CUCEI, Universidad de Guadalajara, {marco.perez, laural.lopez, erik.cuevas, daniel.zaldivar, marco.cedano}@cucei.udg.mx,

Abstract. This work aims to contribute towards developing a novel platform for deploying Visual Servoing (VS) applications over mobile processing units. As several operating systems and programming environments are available within a quite volatile market, one platform must be selected in order to explore its actual capacities for deploying a visually guided robotic application. This paper presents the on-going evolution of one project targeting the development of a real-time VS control toolkit under the Windows7 mobile platform and its related tools. The discussion includes first results along with some conclusions and future work.

Keywords: Windows7 Mobile Programming, Visual Servoing, C# language.

1 Introduction

Recent technology developments have enriched the consumer market by providing powerful processing platforms ranging from mobile telephone to gaming consoles. In particular, the technology around the last generation of mobile phones, largely known as smartphones, have ubiquitously integrated several devices which encompass novel user interfaces, powerful processors and a handy sensor set with relevant telecommunications ports under a unified architecture.

A comprehensive consumer market analysis from 2006 [1], estimated that there were more than 2.5 billion mobile phones worldwide at the time, with numbers at 2009 reaching 175 million of smartphones alone [2]. In the last four years, the development has been accelerated as many companies are investing largely on developing new hardware platforms and software applications. In 2010, the mobile market widened with experienced companies re-launching their efforts either to relocate their technology surpassing competitors in the segment or to introduce more powerful processing devices.

Under such context, mobile processing in general is likely to drive computing subjects for the foreseeable future as new applications and services are becoming available worldwide [3]. Subjects such as distributed computing and pervasive mobile

Received 23/01/11
Accepted 20/02/11
Final version 07/03/11

processing are rapidly evolving and becoming attractive to several subjects on science and engineering.

On the other hand, Visual Servoing (VS) is a research area regarding the commanding of robotic devices from visual information. It naturally shares common issues with robotic control, real-time systems and in particular with computer visión as several visual algorithms are demanded by VS, including active vision, visual pose estimation and dynamic vision.

This work aims to contribute towards developing a novel platform for deploying VS application over mobile processing units. As several operating systems and programming environments are available within a quite volatile market, one platform must be selected in order to explore its actual capacities for deploying a visually guided robotic application. This paper presents the on-going evolution of one project targeting the development of a real-time VS control toolkit under the Windows7 mobile platform and its related tools.

This paper organized as follows: Section 2 provides information about PDA processing platforms while Section 3 depicts a particular discussion on the Windows7 mobile platform including its application development environment. Section 4 presents a quick overview of VS while Section 5 presents the target VS application. Section 6 offers details about some simulation results and Section 7 discusses on some conclusions and the work ahead.

2 Mobile processing platforms

Following a wide heterogeneity on the mobile processing market, developing applications for a given selected platform still possess a clear challenge for any programmer. First of all, cross-platform programming is still at a very early development stage and knowing more than two mobile programming languages up to an expert level, still demands a breath-taking exercise. However, under such circumstances, exploring the use of an integrated processing unit has become extremely attractive to robotics and its control requirements.

A quick overview on the available hardware platforms and its correspondent software libraries is presented by Table 1 as an overview of the information recently presented in [2]. Such table includes a reduced notation as follows: IDE = Integrated development environment, SDK =software development kit, ADT = Android development tools, JDE = Java development environment, PDK = Palm development kit and NDK = native-code development kit.

In general, the Windows phone7 platform has been selected thanks to its mature programming platform based on the Microsoft Visual Studio. Our research group has done a wide effort to develop new libraries and side applications for robotics research on this platform. Just recently, new potential applications are emerging from the Intel's OpenCV library for Windows Mobile 7 which promises to speed up the application development regarding computational vision programming on mobile platforms.

Table 1. Available smartphone platforms: one hardware comparison array

Platform	iOS 4	Android 2.1 with Sense	Symbian^3	WebOS	Windows Mobile 7
Processor	Apple A4	1GHz Qualcomm Snapdragon	680MHz ARM_{11}-based	600MHz TI $OMAP_{3430}$	1GHz Qualcomm Snapdragon
Storage	16GB / 32GB internal	440MB internal, microSDHC expansion	16GB internal, microSDHC expansion	16GB	Approx. 200MB internal,
Cellular	Quadband GSM, pentaband HSPA	CDMA, EV-DO Rev. A, WiMAX	Quadband GSM, pentaband HSPA	CDMA / EV-DO Rev. A	Quadband GSM, dualband HSPA
WiFi	802.11 b/g/n	802.11 b/g	802.11 b/g/n	802.11b/g	802.11 b/g[1]
Display size	3.5 inches	4.3 inches	3.5 inches	3.1 inches	4.3 inches
Display resol	960 x 649	800 x 480	640 x 360	480 x 320	800 x 480
Display tech.	IPS LCD	LCD	AMOLED	LCD	LCD
Primary camera	5 megapixel AF, LED flash	8 megapixel AF, LED flash	12 megapixel AF, xenon flash	3 megapixel, LED flash	5 megapixel AF, LED flash
Video record	720p at 30fps	720p at 24fps	720p at 25fps	VGA 30fps	VGA at 30fps
Location / orientation sensors	AGPS, compass, accelerometer, gyroscope	AGPS, compass, accelerometer	AGPS, compass, accelerometer	AGPS, accelero-meter	AGPS, compass, accelero-meter

From a hardware perspective, accessing the camera image and transferring the information into the work memory has been greatly simplified through the Silverlight© library, allowing new robotics applications to be envisioned at short distance. The energy management hardware has been impressively redesigned with sufficiently long service availability which reinforces its usage for robotic applications. Poor cross-platform code transportation still represent its main drawback as the Windows 7 platform focus entirely on C# programming language, making practically impossible to transport code from other programming platforms. Table 2 shows a handy comparison between commercially available Smartphone platforms. It is easy to browse through their capabilities and services.

3 Microsoft's Windows7 mobile platform

The novel Windows7 mobile operating system, also known as Win7, has evolved to become the second most popular system accessing the Internet worldwide. Its overall usage lies far away from the desktop version of MS Windows, its parental relative.

The Windows7 platform works over an Qualcomm Snapdragon QSD8650 hardware set which includes a GPU, built-in RAM memory and a central processor running over 1 GHz. A hard convention has been agreed regarding the screen size and resolution, considering 800 x 480 pixels.

Table 2. Available programming platforms to date and their correspondent compatible hardware.

OS	Runs on	Related to	Development languages	IDEs, Libraries, frameworks
Android	Open Handset Alliance	Linux	Java	Android SDK and NDK; ADT plug-in for Eclipse
BlackBerry OS	BlackBerry	Unix (BSD and NeXTstep)	Java	BlackBerry JDE
iPhone OS	iPhone, iPad, iPod Touch	Mac OS X	Objective-C / C++	iPhone SDK
PALM WEB OS	Palm Pre, Pixi	Linux	HTML, CSS, Java Script; C / C++ (vía PDK)	PDK; WebOS plug-in DK; Project Ares (Web based)
Symbian OS	ARM processors	Psion EPOC	C++, Java, others	
Windows Mobile	Windows Mobile smartphones	Windows CE	Visual C++	Windows Phone SDK (works with Visual Studio)

4 Windows7 Mobile Application Development

The development under the Win7 standard follows the popular convention of Model-View-Controller which organizes the overall code by separating data from the user interface definition and the required logic control sentences. The first component, known as the Model, aims to validate access to all required data and their functionality. A second component, known as the View, is basically concerned to the user interface while the third component is called the controller because it must reply to common user events that may eventually modify the goal-achievement management of the program.

The integrated development environment which natively serves the Windows7 mobile programming is the MS Visual Studio©. The environment is a friendly-user interface that makes intensive use of graphic tools and manages different platforms under the same environment. A quick view of its classical Integrated Developer Environment can be seen at Fig. 1. The Interface Builder is the code development hosting the visual interface designing and the correspondent user interface. The builder includes a link to the code behind each control in the window. Another important component on the Windows7 mobile environment is the device simulator which is included into the Visual Studio, providing a quick interaction with operating controls and output fields in the software.

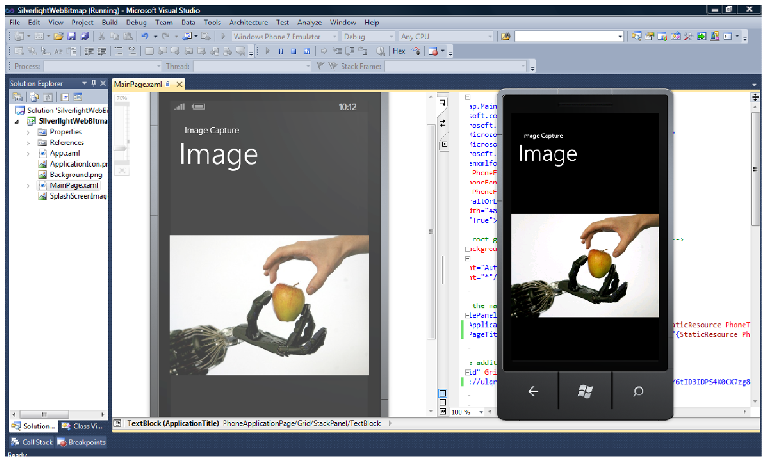
Image Capture
Image
MainPage.xaml
Solution Explorer
TextBlock (ApplicationTitle) PhoneApplicationPage/Grid/StackPanel/TextBlock

Fig. 1: Visual Studio interface builder and related windows.

4.1 Windows 7 programming code

The overall Windows7 programming (WP7) environment has also defined its own development based on the C# language which is built as a C-language meta-set. The Application Program Interface (API) includes high level functions and waste intelligent recollection features. On the other hand, the overall control structure is similar to C language, despite all objects are always created and maintained over a dynamically appointed memory.

WP7 offers transparent interoperability to other Windows-based OS. Therefore, code reusability is assured as the new technology Windows Presentation Foundation is available to assist the transport of desktop applications into the mobile platforms. Despite C# is the main language in the platform WP7 provides an easy bridge to welcome other languages. A code sample for deploying a basic camera application is explained below giving a quick insight into the language structure.

The System.Windows.Media.Imaging library is required in order to access all video-coding functions. The code example below refers to most common camera manipulation functionalities as follows:

```
void btnCapturar_Click(object sender, RoutedEventArgs e)
{
  //Validates if video capture is allowed.
  if (!CaptureDeviceConfiguration.AllowedDeviceAccess )
  {
    CaptureDeviceConfiguration.RequestDeviceAccess();
  }
```

```
            //If such validation is true, it is possible to assign the handler variable.
            //Video declaration (_cs)
            if (CaptureDeviceConfiguration.RequestDeviceAccess())
            {
                System.Windows.Media.VideoCaptureDevice videodev;
                videodev = CaptureDeviceConfiguration.GetDefaultVideoCaptureDevice();
                if (videodev !=null)
                {
                    VideoBrush vb = new VideoBrush();
                    vb.SetSource(_cs);
                    _cs.Start();
                    grid1.Background = vb;
                }
            }
        }
```

5 Target VS application

Visual Servoing (VS) is a mature research area which shares common issues with robotic control, real-time systems and in special with computer vision, involving subjects often employed in VS schemes such as active vision, visual pose estimation and dynamic vision [4]. Different implementations of VS have been traditionally identified within two main classifications: position-based and image-based visual servoing. An introductory overview of both classes is presented in the now classic VS tutorial in [5]. Basically, in the image-based visual servoing (IBVS) the error signal is computed in the image plane and the regulation commands are generated with respect to such error by means of a visual Jacobian. On the other hand, in the position-based schemes, the image features are used to estimate an object-workspace characterization in such a way that the error can be computed in the Cartesian space and used in the control loop.

Traditionally image-based systems have been regarded to possess a good robustness to calibration errors [6], even in the absence of the object and workspace model. However image-based VS schemes also exhibit some weaknesses such as singularities in the visual Jacobian which may lead to conflicts in the control loop. Other major drawback resides in the fact that an image-based system does not control the robot's end-effector in the Cartesian space, sometimes resulting in complicated or even unrealistic joint configurations being demanded to the robot. New IBVS schemes have been proposed to avoid these problems. Some of the new schemes combine 2-D and 3-D information and pose estimation to create a more complete visual servoing algorithm, for instance the 2-1/2 Visual Servoing [6]. However these servoing schemes often require the description of object or its visual features.

In this paper, a low-profile anthropomorphic planar manipulator is equipped with a commercial CCD camera attached to its end-effector. An IBVS algorithm is used to achieve low-speed tracking of a moving object, which is model-free in the sense that no object model is provided [7]. Like other tracking problems, the aim is to keep the camera in a plane parallel to the tracked object. The objective is to investigate the use of mobile processing units to implement computational vision task in a first stage and the overall servoing control scheme in a second stage.

5.1 Experimental Set

The VS platform in this paper, requires a special but easy to conform hardware setup. The mobile telephone unit and one computer are required to implement the visual servoing system. Basically the smartphone performs all the artificial vision processing and computer controls the robot. The first one is known as the Eye Processor (EP) whereas the second one as the Visual Servoing Controller (VSC). Their tasks can be simple enumerated as follows: the EP provides a high level interface with camera drivers and other user interfaces to two visual tracking algorithms, the colour-based and the optic-flow-based trackers [8]. Also it provides the access point to the network communication sub-process which is in charge of establishing the high-speed link to the VSC computer through a reliable filtered link which includes a set of digital filters to overcome noise in the image and in the network link. On the other hand, the VSC computer provides the server for the network link, with an optional input filter to cancel out noise in the network hardware.

The experimental setup employs the low-cost TQ MA2000 6-DOF robot manipulator which has been designed primarily for teaching purposes. It has limited repeatability, low accuracy and sometimes high mechanical backlash. This planar manipulator is equipped with an Win7 mobile phone attached to its end-effector. The W7 phone's built-in camera provides high-quality and low-noise images at a low rate of 30fps. Actually, the W7 mobile phone processor implements the EP duties. As it is discussed later in this paper, it has been envisioned that future experiment with this platform would seek to implement VSC tasks on the same Win7 mobile phone processing unit.

5.2 Camera Calibration

The intrinsic and extrinsic camera parameters for the built-in Win7 mobile phone camera should be experimentally calculated by using the camera model and calibration method from [7]. The Win7 camera model is thus compacted into the classical C matrix:

$$C = \begin{bmatrix} f_u & \alpha_c f_u & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 3198 & 0 & 1218 \\ 0 & 3278 & 767 \\ 0 & 0 & 0 \end{bmatrix} \tag{1.1}$$

With f_u and f_v being the focal distance in pixel units whereas α_c=0, f_c, u_0 and v_0 represent the camera intrinsic parameters. Notice that unfortunately the camera in the mobile handset holds a considerable focal distance which seriously limits its actual capability to focus on closer objects.

5.3 Experiment simulation

The control task can be described as follows: the TQ MA2000 Robot is resting over a flat surface. A circular miniature railway is placed within the robot's workspace, 0.10

meters away from its base. The camera registers four features of one replica train running over the track in order to perform the tracking. The train motion is simulated as a simple circular motion with a constant angular velocity as follows:

$$\begin{bmatrix} x_{train} \\ y_{train} \\ z_{train} \end{bmatrix} = \begin{bmatrix} 0.44\cos(-0.698t - \tfrac{3}{4}\pi) + 0.54 \\ 0.44\sin(-0.698t - \tfrac{3}{4}\pi) \\ 0.04 \end{bmatrix} \tag{1.2}$$

with r = 0.44 meters and the railway circle centred at (0.54, 0, 0). The phase value of 3/4π locates the initial position of the train in the right corner of the robot's workspace. The negative sign in the expression produces a clock-wise rotation. After the initial design and training stage, the VS system is tested on simulation over the MA2000 robot. The performance is analysed through the step-response of the VS tracking system. Initially the robot and the camera are located over the moving object following the teaching-by showing technique. The object is a small replica train. The step response experiment begins with the train moving from its stationary position at a constant velocity of 3.65 cm/sec, following a straight trajectory. As the robot starts tracking, the object stops after 2 seconds allowing the robot to completely catch the object as its features return to the target location in the image space.

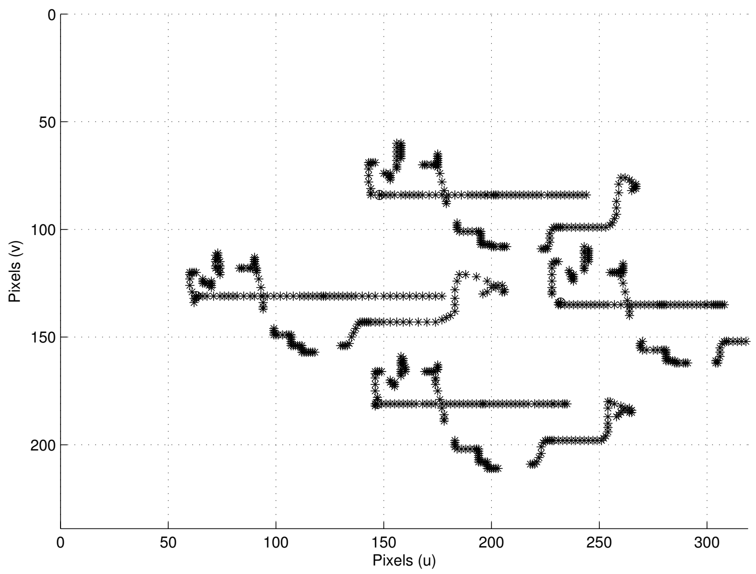

Fig. 2: Trajectory of the features in the image plane with respect to their target location in the step response experiment using linear controllers.

6 Simulation results

Results from the simulation are used by the designer to improve the real-time system performance. The system is simulated for a step response with an operative interval of 2 seconds, which is the time required for the robot to catch the train's movement. Fig. 2 shows the trajectory of the features in the image plane using a PI regulator with K_p = -2.0 and K_i = -0.1 for a step response. Fig. 3 shows the trajectory for each link in the MA2000 robot. Notice how all links go to an stable response after 6 seconds approximately.

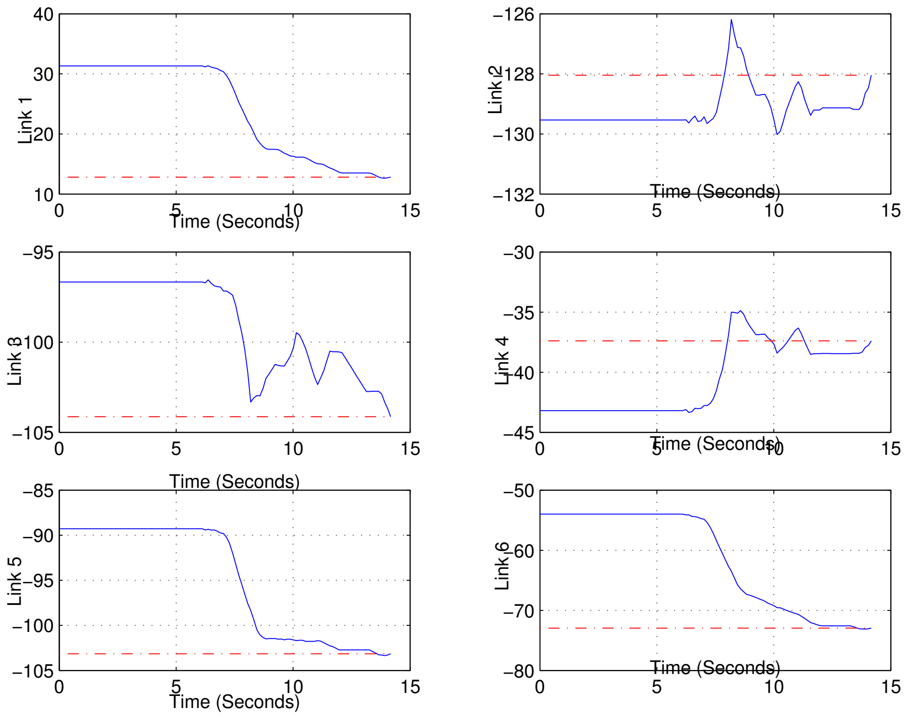

Fig. 3: Robot states during the step response simulation.

7 Conclusions and future work

This paper has presented the on-going research on developing a full set of VS libraries for mobile processing units such as smartphones or similar devices. To date, the work has focused on exploring each available platform in the market, seeking for robust development environment. However, much remains for future implementation such as the real-time operation and the cross-platform implementation of final VS libraries over mobile processing units.

References

1. Lars Kulik, "Mobile Computing Systems Programming: A Graduate Distributed Computing Course", IEEE Distributed Systems Online, vol. 8, no. 5, 2007.
2. Daniel Derns.: Tools & Toys, Writing Small. In: IEEE Spectrum. June 2010.
3. Gregory D. Abowd, Georgia Teach, Liviu Iftode, Helena Mitchell: The Smart Phone: A first Platform for Pervasive Computing. IEEE CS April-June 2005.
4. M A Perez-Cisneros & P A Cook, 2004, Open Platform for Real-Time Robotic Visual Servoing, Procs of 10th IASTED Int Conf on Robotics & Applications Hawaii, August, pp.142-147
5. Francois Chaumette, Seth Hutchinson, "Visual Servo Control Part I: Basic Approaches". IEEE Robotics and Automation Magazine, 13:4 (2006), 82-90.
6. C. Collewet and , F. Chaumette. "Positioning a camera with respect to planar objects of unknown shape by coupling 2-D visual servoing and 3-D estimations". IEEE Transactions on Robotics and Automation, 18(3):322-333, 2002.
7. Heikkilä, Janne and Silven, Olli. "A Four-Step Camera Calibration Procedure with Implicit Image Correction", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 1997, San Juan, Puerto Rico, pp. 1106-1112
8. Cuevas, E., Zaldivar, D. and Perez-Cisneros, M., "Procesamiento Digital de Imágenes con Matlab y Simulink", Book, written on Spanish, ISBN 9785478979738, RA-MA, SPAIN