UNIVERSITÀ DEGLI STUDI DI BRESCIA
FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA INFORMATICA
TESI DI LAUREA SPECIALISTICA

ANALISI DI FATTIBILITÀ ED IMPLEMENTAZIONE DI UN SISTEMA DI ROBOTICA COGNITIVA PER COMPITI DI NAVIGAZIONE

(Analysis of preconditions and implementation of a Cognitive Robotic System for navigation tasks)

Relatore:

Ch.mo Prof. Riccardo Cassinis

Correlatore:

Ch.mo Prof. Marco Ragni

Laureando:
Francesco Bonfadelli
Matricola 83174

Anno Accademico 2012-2013

Contents

List of Figures

1. State Of The Art

1.0.1. ACT-R

1.0.2. OpenCV

OpenCV, an abbreviation that stands for *Open Source Computer Vision*, is a computer vision library that was originally developed by Intel and, later on, by Willow Garage. It is a cross-platform library, released under a BSD license, thus it is free and open source. In the beginning it was developed in C and C++ and afterwards it was expanded by the addition of interfaces for Java and Python. OpenCV is designed for computational efficiency and with a strong focus on real-time applications. The version 2.4 has more than 2500 algorithms. The library has been used in many applications as, for example, mine inspection and robotics [Ope12b]. The following sections contain a brief history of the library and a list of its main features.

History

The OpenCV Project started in 1999 as an Intel Reasearch initiative aimed to improve CPU intensive applications as a part of projects including real-time ray tracing and 3D display walls. The early goals of the project were developing optimized code for basic vision infrastructure, spreading this infrastructure to developers and making it portable and available for free, using a license that let the developers create both commercial and free applications.
The first alpha version was released to the public in 2000, followed by five beta versions between 2001 and 2005, which lead to version 1.0 in 2006. In 2008, the technology incubator Willow Garage begun supporting the project and, in the same year, version 1.1 was released. In October 2009, OpenCV 2.0 was released. It includes many improvements, such as a better C++ interface, more programming patterns, new functions and an optimization for multi-core architectures. According to the current OpenCV release plan, a new version of the library is delivered on a six-months basis. [Ope12a].

Main Features

OpenCV offers a wide range of possibilities. First of all, it provides an easy way to manage image and video data types. It also offers functions to load, copy, edit, convert and store images and a basic graphical user interface that lets the developers handle keyboard and mouse and display images and videos. The library lets manipulate images even with matrix and vector algebra routines. It supports the most common dynamic data structures and offers many different basic image processing functions: filtering, edge and corner detection, color conversion, sampling and interpolation, morphological operations, histograms and image pyramids. Beyond this, it integrates many functions for structural analysis of the image, camera calibration, motion analysis and object recognition. [Aga06].

2. Requirements

2.1. Functional Requirements

The final objective of this work is to make easier the creation of an ACT-R test.
At the moment, a psychologist, in order to create a test, has to create two interfaces, one for the user and one for ACT-R. The user interface is quite simple, it is formed by simple shapes that can have different colours and words. The user has to read or watch the screen and then choose between a series of possibilities. In order to be analysed by ACT-R, this interface must have a corresponding one, which is simpler and contains only the most important features of the first one. This "simpler" interface is given as input to ACT-R and is written according to a fixed pattern. The figure below shows an example of the user interface of a test and of its corresponding one.

The purpose of this work is to create a module which is able to receive as input the user interface, recognize the main features in it and create the corresponding ACT-R interface, which must contain only such features. In order to do this, the software must be made by two parts. The first part will analyze the image. It will recognize and distinguish simple shapes, such as squares, rectangles, circles, stars, arrows; distinguish the colours of the objects; determine the position of each object in the image and recognize the text in the image. The second part will create the correspondent interface for ACT-R, thus...

The software is going to be tested with several ACT-R test.

2.2. Non Functional Requirements

3. Development Process

3.1. SCRUM

3.2. Working Instruments

In addition to ACT-R and OpenCV, many other tools have been used. Here follows a list of the most important ones.

- **Eclipse IDE for C/C++ Developers** as *integrated development environment* [1];
- **Cute** as *unit testing framework* [2];
- **Mylyn** as *task and application lifecycle management* [3];
- **Trac** as *bug tracking system* [4];
- **Git** as *version control system* [5];
- **Dia** and **cpp2dia** to create UML diagrams [6];
- **ZBar** to read QR codes [7].

[1] More informations at www.eclipse.org
[2] More informations at http://cute-test.com
[3] More informations at www.eclipse.org/mylyn
[4] More informations at trac.edgewall.org
[5] More informations at git-scm.com
[6] More informations at dia-installer.de and at cpp2dia.sourceforge.net
[7] More informations at zbar.sourceforge.net

4. Project

5. Implementation

References

[Aga06] Gady Agam. Introduction to programming with OpenCV. Technical report, 2006.

[Ope12a] Opencv change logs, 2012. Available on line.

[Ope12b] Opencv web page, 2012. Available on line.