Customer: Kjell Lemström Instructor: Anita Kasari

OT4: pics99

Design Requirements

Timo Immonen
Jussi Lindgren
Sampo Pyysalo

Helsinki March 10, 1999

Design Requirements
Software Engineering Project
UNIVERSITY OF HELSINKI
Department of Computer Science

Contents

Version Number	Comments	Date/Author
0.1	chapter 6(?), first versions	1.30.99 /JL
0.5	tentative versions of chap. 1-2	2.10.99 /TI
0.6	chapters 1-2	2.15.99 /TI
0.7	references	2.15.99 /TI
1.1	corrected (own) errors	2.18.99 /TI
1.3	changes per meeting	2.23.99 /TI
1.4.	changes per meeting	3.10.99 /TI

1 Introduction (TI)

PICSearch is a development tool for content-based image retrieval algorithms. It is aimed to serve as a testing platform for new retrieval algorithms. It can also be used as a conventinal image cataloging program.

The initial version PICSearch was developed at Helsinki University of Technology. First version, **PICSearch v1.0**[1], was restricted to use a local image database only. Further development at University of Helsinki resulted in **PICSearch v2.0**[2], a version enhanced with rudimentary network capabilities. A third version of program was also developed at University of Helsinki. **PICSearch v3.0**[3] added a cache implementation for greater network efficiency and also featured an improved graphical user interface.

The pics99 Software Engineering Project group will continue to develop the previous version of PICSearch program, **PICSearch v3.0**, by systematically debugging and optimizing existing features and by adding further capabilities to the program. The result of the project will be a new version of the program, **PICSearch v3.1**.

This document will clarify decisions made by project group and customer considering the aims of the pics99 project. In section 2 existing PICSearch program is introduced. Section 3 establishes general guidelines of the project, sections 4 and 5 discuss the first and the second phase of the project respectively, while section 6 covers other code-related issues. Testing is discussed in section 7, and implementation in section 8. Finally, limitations of the project are considered in section 9.

2 Existing PICSearch Program (TI)

Current PICSearch program is an evolutionary product composed of the results of several preceding projects. As such it has become impossible to document the program coherently and completely without major effort. Readers are therefore advised to consult design requirements of previous projects, if specific implementation choices or general design questions need to be clarified [4] [5] [6].

This document will not, however, make use of any terms or component names defined in previous documents, but will examine closer those parts of the program that pics99 project team will propose to modify.

3 General Guidelines of the Project (TI)

3.1 Project Motivation and Intended Users

PICSearch was originally strongly oriented towards non-casual users. Persons interested in content-based image retrieval algorithms were considered to be motivated enough to use even a slightly cumbersome program. Therefore user interface and ease-of-use of the program were not set high priority on initial requirements, or were somewhat neglected during implementation.

Successive projects have improved usability and have tried to make PICSearch more applicable as a conventional image database. However, those projects have also introduced many bugs and faulty implementations along with their new features and capabilities. In general those measures taken to create a feasible conventional image database out of PICSearch system have been inadequate. Furthermore, we strongly feel that focus of the PICSearch program should be professional and semi-professional users. This project will therefore aim to produce a more usable research tool instead of a multirole image database system.

3.2 Structure of the Project

Because several preceding projects have somewhat cluttered the implementation code of the PICSearch program, pics99 project group has decided to proceed with the project in two distinct phases: First phase is devoted to debugging and removing "dead code" and aims to produce a more compact and stable program structure. This is considered desirable for reasons of usability and further development potential of the PICSearch software.

Second phase will let the group to take advantage of familiarity with PICSearch code inevitably gained by debugging and optimizing done during the first phase of the project. Product of the second project phase will be a major modification of the program. The modification will make adding new algorithms to PICSearch system easier by separating all algorithm operations to independent processes communicating with PICSearch server process.

4 First Phase of the Project(SP)

4.1 Overview

The objective of the first phase is to improve the current program without significantly modifying its functionality. Although the emphasis will be on debugging and restucturing the code, some of the more obvious deficiencies in the program will also be addressed. The first phase will result in a more stable and maintainable program while giving the project team a better undestanding of its inner workings.

4.2 Planned improvements

Because of the general nature of this document the problems associated with the code structure and implementation details will not be discussed here. Instead, a brief overview of the more visible changes that have been planned for the first phase follows:

1. The client does not currently handle "cancel query" requests correctly but hangs instead. This will be fixed so that the client will not hang.
2. The client has a history list of the last five servers it was connected to. As the current implementation does not eliminate duplicates from this list it can easily contain five "local host" entries. The possibility of duplicates appearing in the list will be eliminated in the first phase.
3. The client hangs if the server provides it with fever thumbnails (as a reply to a query) than it wishes to display at a time. This will be fixed so that the client will not hang.
4. The "Query result images" and "Minimum match percentage" client-side parameters are ignored. These parameters will be put to use.
5. The "IDSearch" and "Name" algorithms do not function correctly. These algorithms will either be removed or fixed so that they return the correct results.
6. The UI is somewhat clumsy. Minor tweaks are in order and will be made if there is enough time.
7. The program lacks a comprehensive help and the "info" menu provides inadequate information at best. If there is time, a more thorough help will be provided.

8. The program exhibits unexpected behaviour when users with superuser rights are added to the userbase. This will be fixed so that superusers can be added correctly.

The visible first phase fixes will include but not be restricted to those mentioned above. In addition to some minor features that were omitted from the list, all previously unknown problems that come up during the first phase will also be fixed.

5 Second Phase of the Project (JL)

5.1 Overview

As PICSearch is intended to be used as an algorithm test platform, it should be as easy as possible to include, modify and remove algorithms. However, these operations are difficult to perform in **PICSearch v3.0** because of the following:

1. Extensive study of the source code is required, because the documentation is unclear on the subject.
2. Both client and server parts have to be shut down in order to do any algorithm maintenance.
3. Although the idea was to concentrate all heavy calculation to the server, the algorithms are used in the client part as well.

To overcome these problems, support for external algorithms will be developed as the product of the second phase. Each algorithm will be its own executable, launched by the server when needed.

This approach will concentrate the calculations to the server, as the algorithm code is no longer executed in the client.

In addition, algorithm maintenance will no longer require study of PICSearch source code, because the algorithms are clearly an external, independent part. As icing on the cake, client and server need no longer to be shut down while modifying algorithms, as server can be told of the changes while it is running.

This modification causes very little visible difference to the end user.

5.2 Services of the External Algorithms

External search algorithms perform only two kinds of operations:

1. Calculating a signature for a picture

   Input: picture data
   Output: signature
   Function:

   Algorithm receives picture data as input. Each algorithm internally decides what kind of information it derives from the data, and then produces and outputs a signature accordingly. The signature has only meaning for the algorithm which produced it.

2. Comparing two signatures

   Input: two signatures, parameters(optional)
   Output: match percentage
   Function:

   Algorithm receives two signatures as input. It compares them according to the given parameters (if any), and outputs a match percentage describing how well the signatures (thus, the original pictures) seem to resemble each other.

6 Other Issues (JL)

6.1 Performance

Database, message handling and graphical user interface of **PICSearch v3.0** seem inefficient. The first phase of the project will not degrade the performance of the system, and in certain areas the performance will get better because of the fine-tuning.

The external algorithm support of the second phase will slow down the signature calculation and comparing with a constant factor.

6.2 Maintainability

As bugs and unused code are removed from the system, it will be easier to maintain. External algorithm support works towards that end as well.

Added functionality will be documented, and the source code on that part will contain commentation. Corrected problems and small improvements will be documented.

7 Testing Procedures (TI)

The first phase operations of the project will be tested as required by guidelines in the project plan [7]. Individual code changes will be evaluated and tested by the person responsible for the code module in question. All operations made during the first phase of the project will be collected to a single modification document.

The second phase product will be tested mainly by integration testing of the whole **PICSearch v3.1**. The template of the algorithm executables will be tested by module testing to verify that its interface functions correctly.

The overall goal of testing is to provide as error-free implementation as possible. Specific requirement on behalf of the customer is that the functionality and usability of the PICSearch program should not be degraded by this project. Comparative testing will be conducted to ensure this.

8 Implementation Enviroment (SP)

PICSearch is developed for Linux/Unix enviroments using C++ (as defined by the December 1996 ISO/ANSI Draft, including the Standard Template Library) and Tcl/tk (version 8.0 of the library). The first version of PICSearch aimed to sustain portability for Windows or other enviroments. In reality at least version 3.0 has compromised portability, and pics99 project will accept further loss of portability if some efficient solutions require it. To ensure that portability is not unnecessarily compromised, POSIX-compliant features will be used whenever possible.

9 Limitations of the Project (TI)

The pics99 project is limited mainly by project timetable. An additional self-imposed limitation is the desire of the project group to make sure that resulting **PICSearch v3.1** is fundamentally more sound program than the previous version **PICSearch v3.0**. In practise this means that more time is allocated during the project to making sure that no additional bugs or performance problems are introduced in the new version. The group will not undertake any modifications it cannot satisfactorily bring to conclusion, even if this will limit the scope of the new features.

Glossary

Following terms have been used in this document. Some meanings may be project-specific.

algorithm

In this document, algorithm references to a content-based image comparing algorithm. Basically, these algorithms just decide how closely two pictures resemble each other. PICSearch uses these to compare a user-provided example image with the pictures in the database.

bug

An error in the source code of a program. A bug is generally held to be an error which does not stop the program from being compiled, but instead causes it to function erratically, stop functioning, or to function in ways not considered appropriate.

dead code

A useless section of the source code of a program, ie. a section which does not get executed when program is run, or which duplicates operations already implemented in some other code section.

debugging

Removing errors (see: bug) from the source code of a program.

signature

Every algorithm produces a signature out of a source image. The signature contains those things the algorithm thinks important in the source image. Signatures of two different pictures are used to check for resemblance, not the pictures themselves.

References

[1]
K. Lemström, J. Korte, P. Kuusi, P. Kyheröinen and P. Päiväkumpu:
PICSearch - A platform for Image Content-based Searching Algorithms, (Technical Report C-1997-48), proceedings of the Sixth International Conference in Central Europe on Computer Graphics and Visualisation 98 (WSCG'98), edited by Vaclav Skala, Chzech Republic, February 9-13, 1998, pp. 222-229.
`http://www.cs.helsinki.fi/ klemstro/publications.html`

[2]
P. Kuusi, K. Lemström, P. Kanerva, S. Koponen, S. Männynsalo, J. Riihimäki, J. Teinilä and T. Virtanen:
A Client-Server Tool for Image Retrieval Applications, In: Proceedings of the HeCSE Workshop on Emerging Technologies in Distributed Systems, Helsinki University of Technology, Department of Computer Science and Engineering, pp. 45-46 (Research Report A50). Otaniemi, Finland. March 1998.

[3]
S. Kaski, T. Koskinen, J. Leinonen and M. Pennannen:
pics98, Implementation Specification, Helsinki, Finland, September 10, 1998.
`http://www.cs.helsinki.fi/group/pics98/`

[4]
J. Korte, P. Kuusi, P. Kyheröinen ja P. Päiväkumpu:
PICSearch-projektin (PICSearch v1) vaatimusmäärittely, Teknillinen korkeakoulu, 1996.

[5]
P. Kanerva, S. Koponen, S. Männynsalo, J. Riihimäki, J. Teinilä ja T. Virtanen:
NetPICSearch-projektin (PICSearch v2) vaatimusmäärittely, Helsingin yliopisto, Tietojenkäsittelytieteen laitos, 1997.
`http://www.cs.helsinki.fi/group/netpics/VM/VaatM12.html`

[6]
S. Kaski, T. Koskinen, J. Leinonen ja M. Pennannen:
pics98, Vaatimusmäärittely, Helsingin yliopisto, Tietojenkäsittelytieteen laitos, 1997.
`http://www.cs.helsinki.fi/group/pics98/`

[7]
T. Immonen, J. Lindgren ja S.Pyysalo:
pics99, Projektisuunnitelma, Helsingin yliopisto, Tietojenkäsittelytieteen laitos, 1999.
`http://www.cs.helsinki.fi/group/pics99/`