QGAR Environment

General Presentation, Perspectives and Discussion

Philippe Dosch

Philippe.Dosch@loria.fr

Philippe.Dosch@loria.fr

loria
Laboratoire lorrain de recherche
en informatique et ses applications

IPOL

June 2012

Summary

Summary

QGAR Purposes

- Collect and capitalize the works of many researchers
- Ease the development of new applications by regrouping state-of-art implementations of basic structures and graphics precessing algorithms
- Provide an environment to tune applications and evaluate their performances
- Spread our know-how in the field of graphics recognition

Architecture
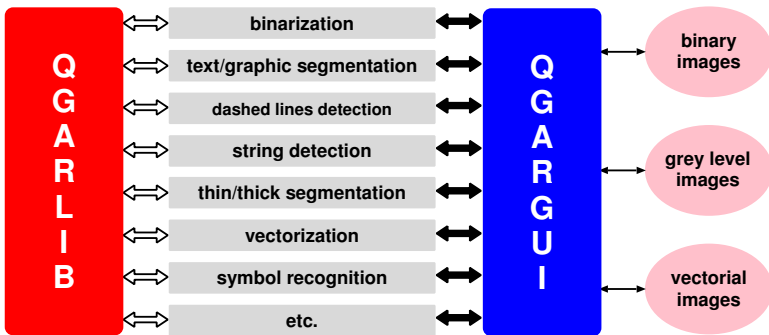
QGAR Library

- Approx. 150 classes, written in C++
- *Image processing*
  Binarizations, mathematical morphology, distance transformations, skeletonization, convolutions, Gradients and Laplacians, edges detection...
- *Graphical processing*
  Polygonal approximations, Freeman chains, connected components, vectorization...
- *Data structures*
  Images, graphs, trees, histograms, masks...
- *Tools*
  Files input/output, object serialization, classification...

QGAR Applications

- Applications built from the basic building blocks from the QGAR Library
- Independent applications
    - Interactive call from the QGAR graphical user interface
    - Batch call from the command line
- Around 10 applications are available
  binarizations, text-graphic separations, thin-thick separations, text extraction, vectorizations, image degradation, symbol recognition...

QGAR GUI

- User interface
    - Process invocation and parameters tuning
    - Results visualization
    - Interactive image editing

- Application import through a plug-in architecture
- Data transfer using files formats PBM+, DXF, SVG
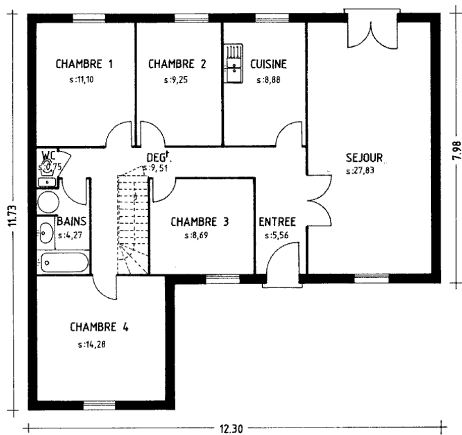
Software Features

- About 170,000 lines of C++
- Unit testing CPP Unit
- Available under Linux/Windows
- Computations not distributed
- Registered as free software by the French agency for software protection (APP)
- Licensed under LGPL/QPL
- Website http://www.qgar.org

Summary

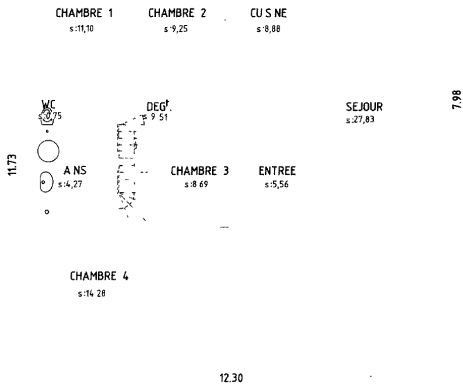**Philippe Dosch**   **QGAR Environment**
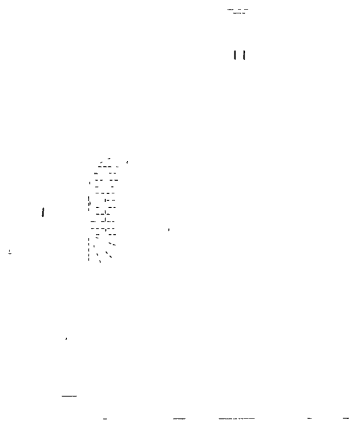
Demo
Original image

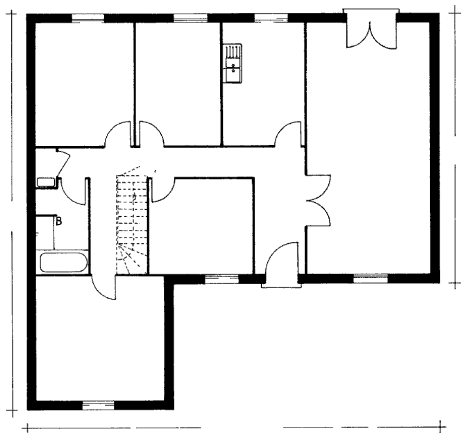# Demo
Text-graphic separation: Text layer

Demo
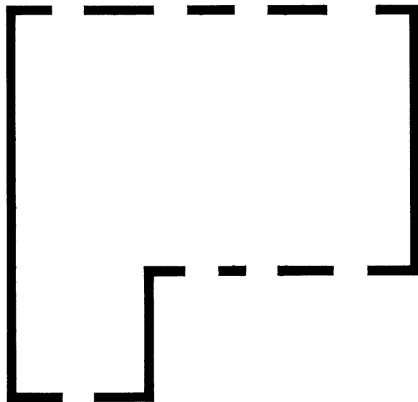Text-graphic separation: Undetermined layer

Demo
Text-graphic separation: Graphic layer

Demo
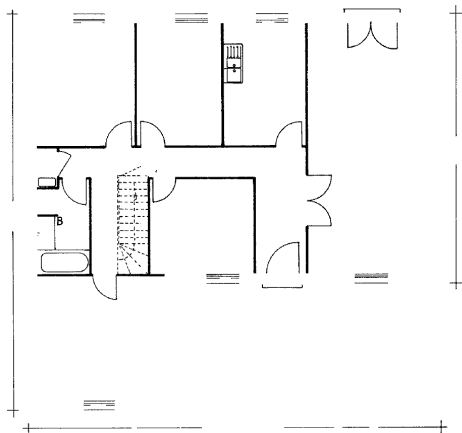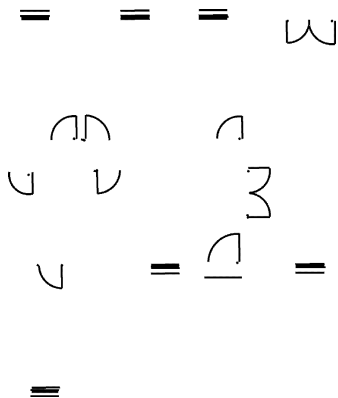Thick-thin separation: Thick layer

Demo
Thick-thin separation: Thin layer

Demo
Recognized symbols
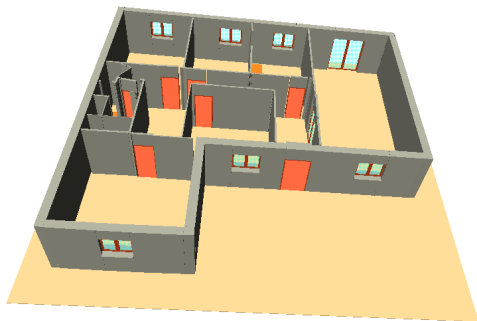
Demo
3D Reconstruction

Summary

Philippe Dosch **QGAR Environment**

A Brief History

- Project started in 2000 when
  - There was a need of works capitalization
  - No such open source environment was available for our needs
  - Half a dozen people were motivated for this environment creation
- The great period: 2000–2005
  - Many contributors: students, researchers, engineers
  - Many users
  - Used in several research projects and industrial contracts

A Brief History

- The hard times: 2006–now
    - Evolutions (internal, external) are time-consuming
    - No "permanent" maintainer
    - New students use to work with Mathlab: less users, less contributors for our environment
    - Even if...
        - Mathlab (or other general scientific softs) does not address all our specific needs (but our environment does?)
        - The environment is still functional (but for how long?)
        - Still provide a (good) visibility of our know-how and generate partners contacts

- And now?

The future?

- IMHO, we have less and less time to devote to research
- Software engineering require time and skills
- A critical number of users/contributors is required to make live such an environment
- Since 2000, new softwares and tools are available: not "perfect", but useful
- So, the future?

The future?

- Devote efforts and time to our environment? How?
- Use as much as possible standard tools. Great, and for specific purposes?
- Use one or several of the many great environments presented today? Each additional use of these could become a problem (license, bug fixes, perenity...), but why not as...
- ...We could also work on a common environment, involving several teams (but also needs...)
- No final answer for now... Charon (presented this morning), for technical aspects?