Human-Machine-Interaction

„About Portable Keyboards with Design and Implementation of a Prototype Using Image Processing“

Presentation of the Semester Thesis
within the Context of
„Human-Machine-Interaction“ (M. Sc.)

Department of Computer Science / Mathematics
Munich University of Applied Sciences

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

fachhochschule münchen
munich university of applied sciences

Contents

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

fachhochschule münchen
munich university of applied sciences

Contents - Introduction

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

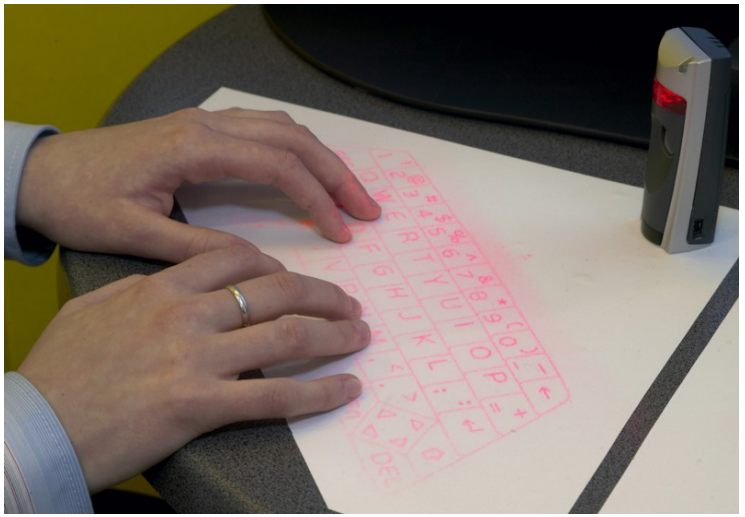
Introduction

Keyboards ...

- are one of the oldest and most-used input devices in existance.
- have always been a vital part of Human-Machine-Interaction.

Tasks

- Provide an overview of the latest developments in portable keyboards.
- Take a detailed look into the projection keyboard *Celluon* [3] (by Canesta Inc. [2]).
- Implement the prototype of a virtual keyboard.



Celluon Projection Keyboard (Courtesy of Canesta Inc.)

Motivation

- Big advances have been made in developing smaller portable devices (i.e. PDA).
- Today's keyboards still look like the first one developed decades ago.

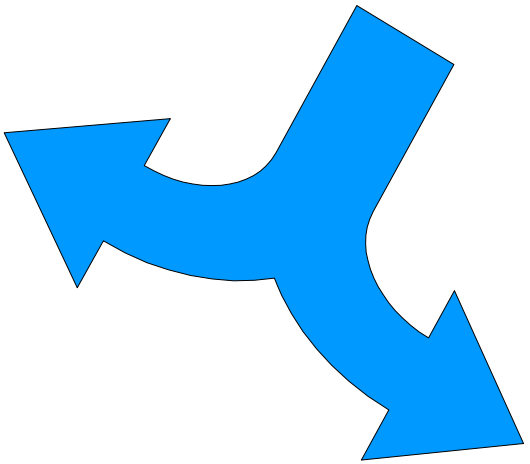
Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Introduction cont.

SGH-D500 cellphone
(Courtesy of Samsung)

Question

Why is there such an obvious development gap between portables like cellphones, and keyboards which are necessary input devices for any computer ?



Keyboard, Push-Buttons, and Mouse, 1960s at Stanford Research Institute [4]

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Contents – Latest Developments

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

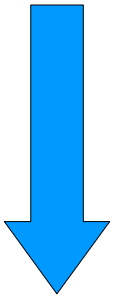
Latest Developments

Categories of today's Industrial Products

- Products keeping the look and layout of the known keyboard.
- Products giving up this nature to gain advantages (portability, usability, etc.).

List of Examples (of course incomplete)

- Foldable Keyboards
- Miniature Keyboards
- Touchscreens
- Glove Keyboard



Changed look and layout compared to common keyboards ...

Specific Example – Projection Keyboard *Celluon*

- Technical Data
- Applications

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Foldable Keyboards

Advantages

- Keeps normal keyboard dimension and layout.
- Ability to use like the „keyboard at home“ without additional learning time.
- Could be included in bags etc. .
- Smaller versions are in experimental use on jackets and other clothing.

Bluetooth textile keyboard
(Courtesy of Eleksen Ltd.) [6]

Disadvantages

- Takes up some space even if folded.
- Depending on the size, it needs a large plain surface to be unfolded.

Textile keyboard
(Courtesy of Eleksen Ltd.) [6]

Example by Eleksen Ltd. [5]

- Rollable textile keyboard.
- Connects to the host device via bluetooth.
- Includes a small battery pack (~ 10 hours [6]).

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Miniature Keyboards

Advantages

- Increased portability due to small size.
- Keeps most keys (depending on actual size).

Disadvantages

- Changed keyboard layout to get accustomed to.
- Extensive work (i.e. typing much text) is not really possible -> Usability suffers from small keys.

Bluetooth Freedom Mini Keyboard
(Courtesy of Hama) [8]

Example by Hama GmbH & Co. KG [7]

- Miniature keyboard offering thirtynine keys.
- Connects to the host device via bluetooth.

Bluetooth Freedom Mini Keyboard
(with cellphone) (Courtesy of Hama) [8]

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Touchscreens

Advantages

- Usage is more intuitive (directly pressing onto the screen).
- Can adapt to the current program situation by hiding/showing user interface elements.
- Also usefull in small sizes depending on the application.

Touchscreen Garmin *StreetPilot 2660* [9]

Disadvantages

- Depending on the size, only a small amount of elements can be visible at one time.
- Offering a full keyboard layout severly reduces the space on the screen.
- Bigger touchscreens are not portable anymore.

Touchscreen Kiosk Mode – *Internet Browsing Software* [10]

Example by Garmin

- Designed to be used in embedded areas (i.e. cars).
- Due to screen size only usable for a limited number of graphical elements.

Example by Kiosk Mode

- To be used in public internet terminals.
- Offers a browser and email client.
- Featuring a full keyboard layout.

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

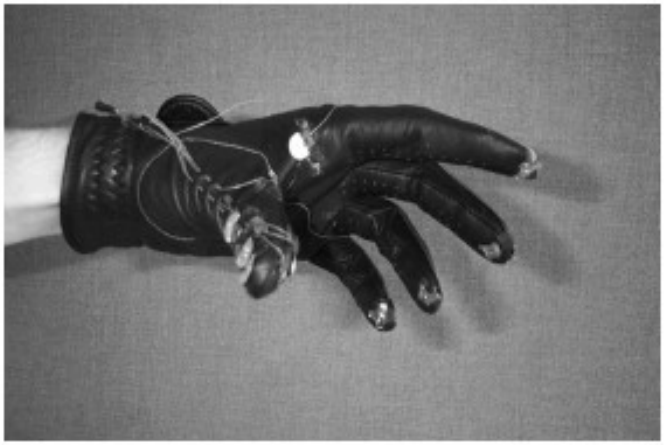
Glove Keyboard

Advantages

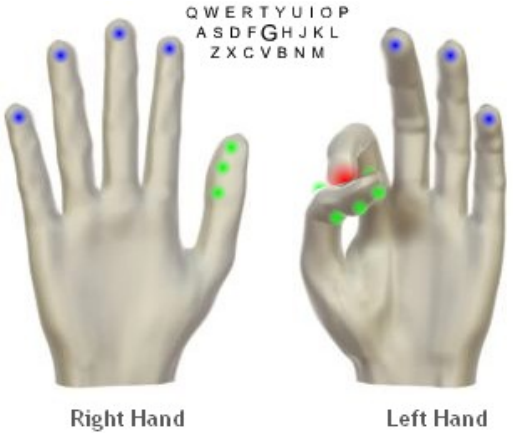
- Optimal portability (can be used while walking).
- Does not need a plain surface.
- High input rate (if usage is mastered).

Disadvantages

- Need to learn a completely different input device.
- Need to wear gloves covering all fingers.



KITTY Glove Prototype
(Courtesy of KITTY Tech.) [11]



KITTY usage Example
(Courtesy of KITTY Tech.) [11]

Example by KITTY Technologies

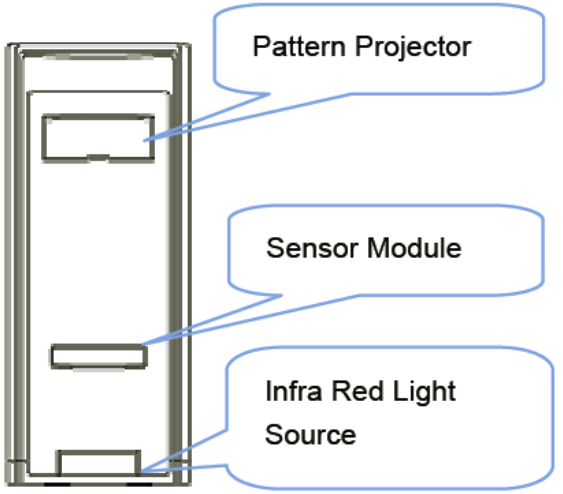
- Follows same idea as old keyboards (connect two wires).
- Connect two fingers at pre-defined areas to generate a key stroke.
- Still under development.

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Celluon Projection Keyboard

Facts

- Replaces the keyboard by a projected image.
- Connects to its host device via bluetooth or serial adapter.



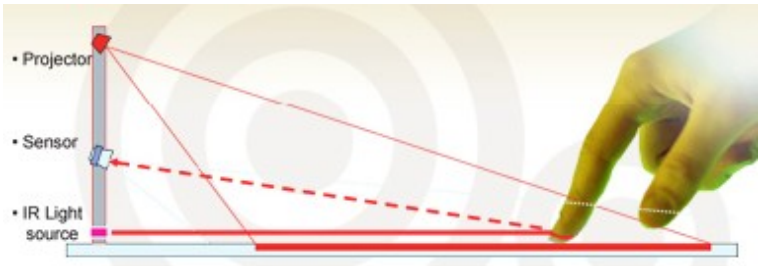
Celluon Tower

Technical Data

- *Pattern Projector*, to display the keyboard layout.
- *Sensor Module*, to detect key strokes.
- *IR Light Source*, to support the detection process.

Usage

- The users' finger is crossing into the threshold IR light.
- Crossing into the light is detected by the sensor.
- The sensor determines which key has been hit.



Celluon usage Example

(all Pictures Courtesy of Canesta Inc.) [3]

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Celluon Projection Keyboard cont.

Advantages

- Fast detection process (no image processing).
- Normal keyboard size and layout.

Disadvantages

- Replacing keyboard layouts for different countries is difficult.
- Battery package limited to 220 minutes max.
- Visibility of the projected image might be a problem under certain light conditions.
- Needs a plain and non-reflective surface.

Siemens *New Interactive Phone*
(Courtesy of Siemens) [12]

Celluon usage Example
(Courtesy of Canesta Inc.) [3]

Applications

- Using the celluon tower at home or at work.
- Incorporate Celluon into existing portable devices (i.e. cellphones).

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

fachhochschule münchen
munich university of applied sciences

Contents - Implementation

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Technical Basics

Preface

- Same approach as used in *Celluon* could not be matched (due to the needed *Pattern Projector* and algorithms).
- New hardware components took the roles of the *Pattern Projector* and *Sensor Module* of the *Celluon.*

Problems

- Projection – solved by replacing the image with a printed paper keyboard.
- Detection – solved by splitting the problem into two individual areas.

Detection

- Threshold detection: „When did the user hit a key?"
- Overview detection: „Which key did the user hit?"

Camera Hardware

Two Logitech *QuickCam Express* webcams:

Resolution	max. 640x480 pixel
Colorsystem	RGB or YUV
Framerate	up to 30 frames per second

Logitech *QuickCam Express*
(Courtesy of Logitech) [13]

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

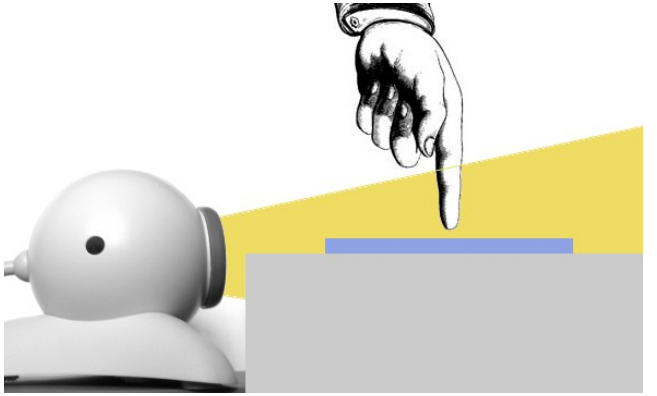
Technical Basics cont.

Threshold Detection

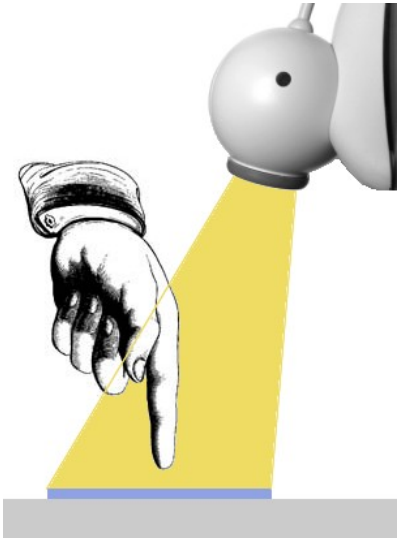
Q: „When did the user hit a key?"

A: The moment the user puts his/her finger down onto the surface.

To detect the height of the finger above the surface a camera must be placed on one side of the keyboard. An alarm is generated when the user enters a pre-defined area above the keyboard, which would count as „key pressed".



Threshold Detection – Camera Placement



Overview Detection
Camera Placement

Overview Detection

Q: „Which key did the user hit?"

A: The moment a threshold alarm is generated it must be evaluated, which key the user has „pressed".

Using the threshold camera for this task was not possible due to the angle needed for a good overview image. A certain key stroke is generated when the user's finger is within a certain pre-defined key area.

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Camera Setup

Problem
Running both cameras at the same time using *Windows XP* and *Java 6.0* [15].

Microsoft Windows XP
XP was able to detect both USB cameras, but unable to run them at the same time. Only different cameras can run at the same time (tested with another webcam).

Solution: Running both cameras simultaneously was not necessary.

1. The threshold detection is running at startup.
2. A threshold alarm is generated.
3. The threshold camera is disabled, enabling the overview camera.
4. A single overview picture is taken to evaluate which key the user hit.
5. The threshold camera is re-enabled.

Sun Java Virtual Machine
JDK [14] is unable to access webcams and needs Sun's Java Media Framework (JMF) [16].

Accessing two webcams was not possible because JMF is using an older Video for Windows (VfW) implementation.

Solution: Using DirectX via a custom DirectShow interface for Java called Java Media DirectShow (JMDS) [17].

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Requirements Analysis

Hardware Configuration
Each camera had to be configured individually for its specific detection task.

Common for both Detections (Settings)
- Choosing the correct camera.
- Setting the camera format (colors, resolution, framerate, etc.).
- Adjust the detection parameters (determined by the image processing algorithm).
- Enable/Disable the detection.

Threshold Detection
- Lower/Upper detection height threshold.
- Show/Hide visualization of the detection area.

Overview Detection
- Management of the detection areas acting as keys (create, delete, etc.).
- Show/Hide visualization of the detection areas.

User Interface
Configuration and detection should be possible via a graphical user interface.

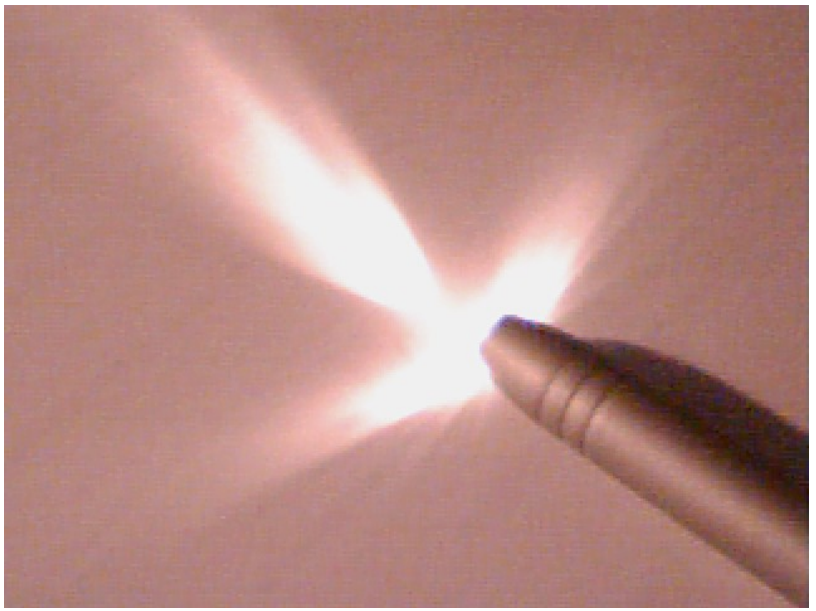
Common for both Detections (UI)
- Opening a main frame supporting a Multiple Document Interface (MDI).
- Menubar and toolbar to open the different configuration and detection dialogs.
- Live-preview of recorded images.
- Possibility to save/load configurations.

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Image Processing Algorithm

Requirement
Easy and fast detection within acceptable framerates (~2 fps+).

Ease-up the Detection
To make the detection more easy and fast, it was decided to focus solely on the color/brightness of a certain object (instead of detecting the user's real finger).



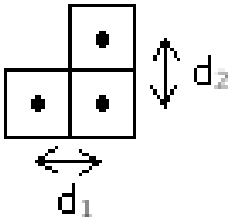
Webcam Picture of the Light-pen used for Detection

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Blob Coloring Algorithm

Goal [19]
Find regions of a pre-defined color/brightness within an image.

Usage
A „backwards L" shaped template is passed over the whole image from left to right and top to bottom.



„backwards L" shaped Template

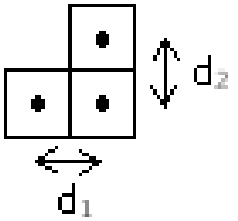
For each pixel calculate the distance ...
- d_1 between itself and its left neighbour.
- d_2 between itself and its upper neighbour.

Definition: „Distance of two pixels"
- Grayscale: difference between the gray levels.
- RGB: Euclidean distance E_{RGB} in the RGB color space.
- HSI: difference between hue or intensity.

$$E_{RGB} = \sqrt{(r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2}$$

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Blob Coloring Algorithm cont.



„backwards L" shaped Template

Result Interpretation

A pixel is considered to belong to a different region if the distance d_i between the adjacent pixel is greater than a certain threshold T.

(d_1 > T) and (d_2 > T)

Pixel is different from both neighbours => assign to a new region.

(d_1 < T) and (d_2 > T)

Pixel is different from above neighbour, but similar to left neighbour => assign to same region as left neighbour.

(d_1 > T) and (d_2 < T)

Pixel is different from left neighbour, but similar to above neighbour => assign to same region as above neighbour.

(d_1 < T) and (d_2 < T)

Pixel is similar to both neighbours => assign it to the same region as the neighbours.

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

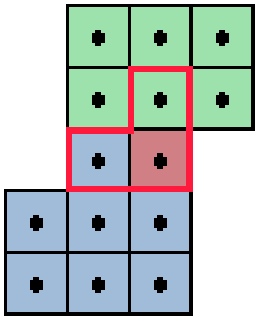
Blob Coloring Algorithm cont.

Problem
Case 4 is problematic:
Current pixel is similar to both neighbours, but the regions for the neighbours differ.

Both neighbour regions differ but are equivalent due to the current pixel connecting both.

Currently examined template	=> „red line“
Current Pixel	=> „red“
Pixels in region „1“	=> „green“
Pixels in region „2“	=> „blue“.

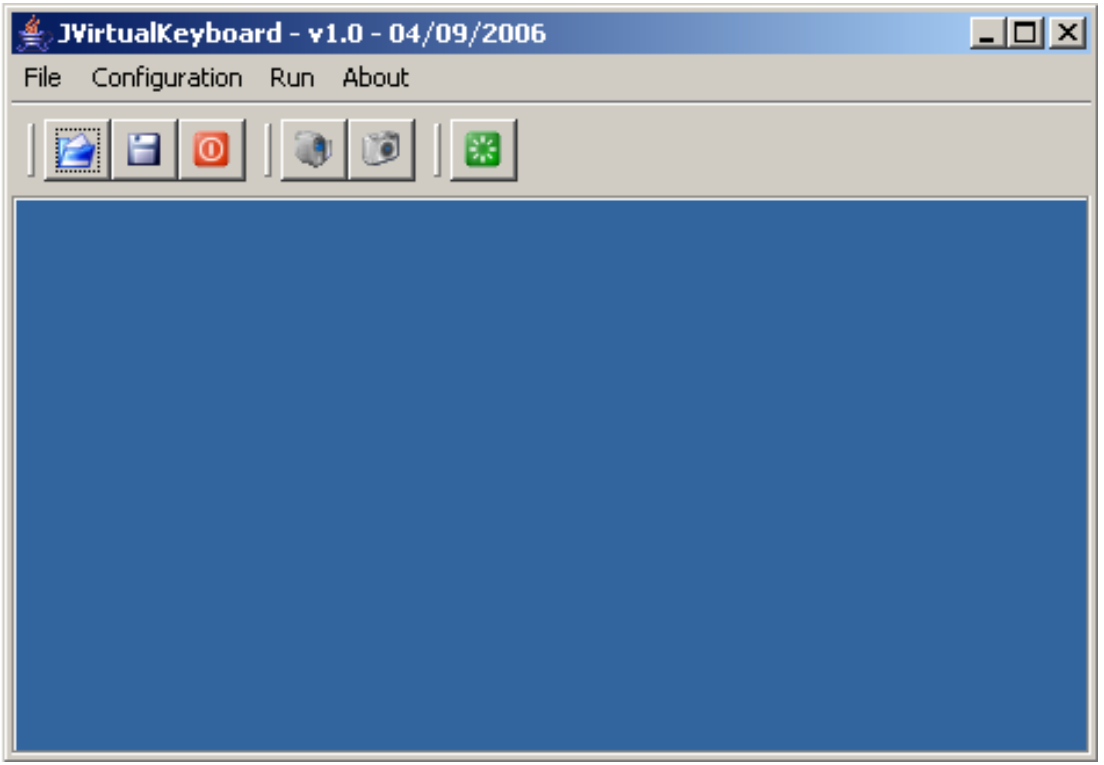
A solution for this problem is presented later during the implementation.



Equivalent Region Problem

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

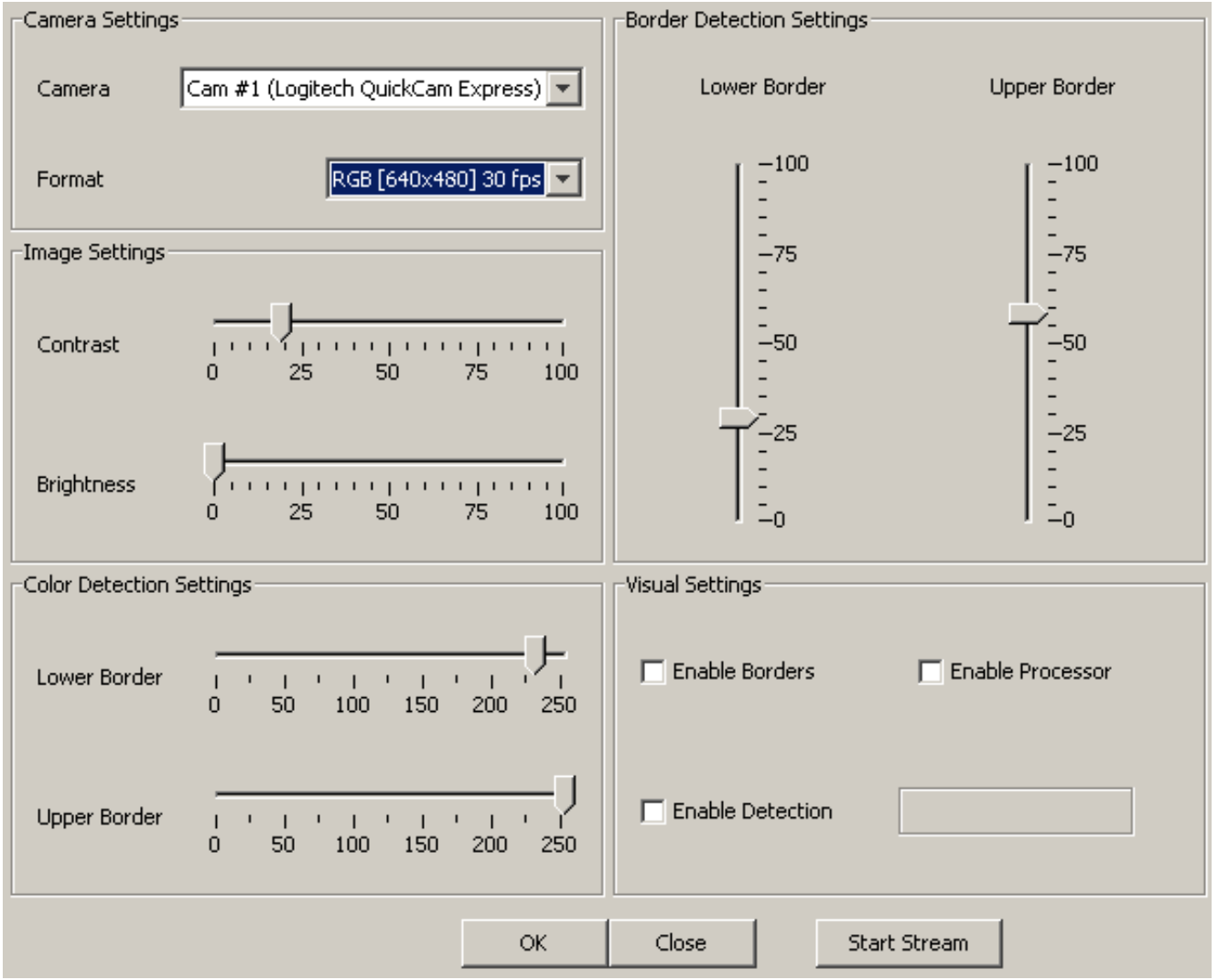
User Interface



Main Window with MDI area

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

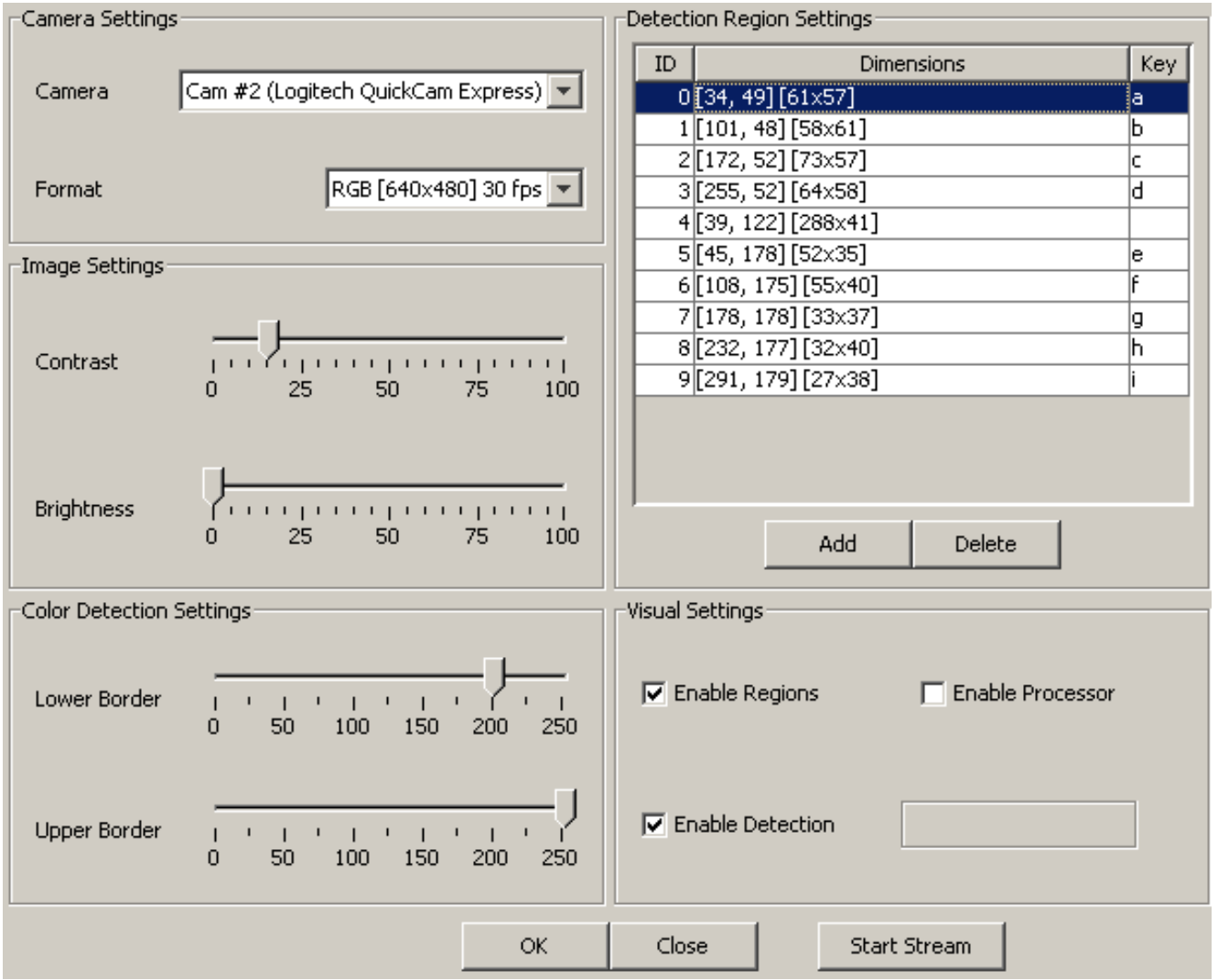
User Interface cont.



Configuration – Threshold Detection

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

User Interface cont.

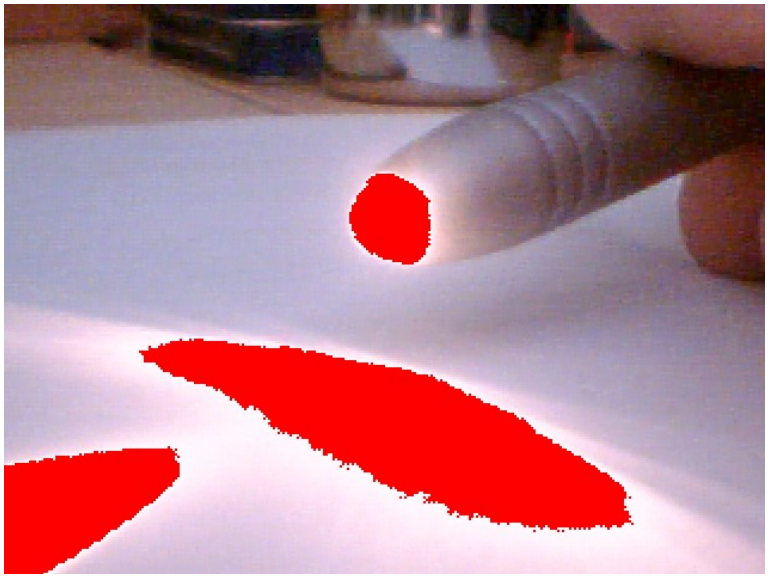


Configuration – Overview Detection

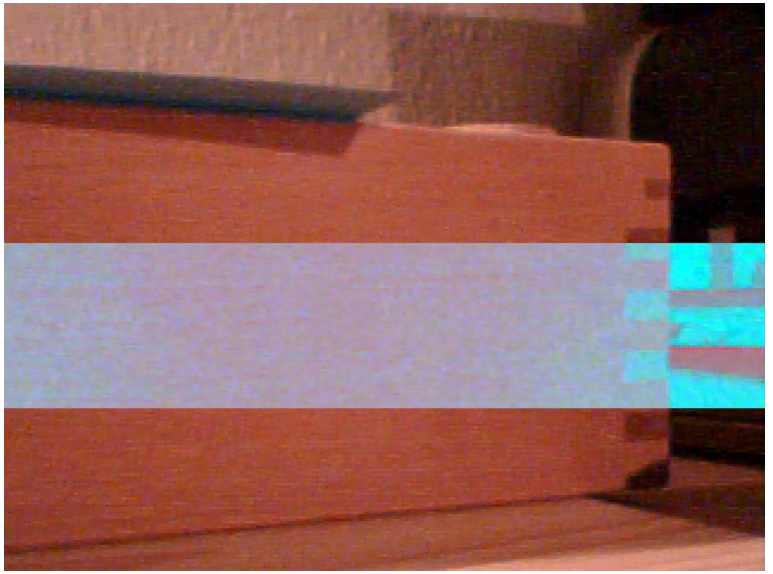
Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Image Workers

ThresholdImageWorker



BorderImageWorker



Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Blob Coloring Algorithm

Common

Pre-processing: All pixels not belonging to the defined color/brightness range are removed from the image (=> color set to „black").

Only colors which we are searching for are now present. Therefore the actual distance does not need to be calculated anymore.
=> It is either „0" (color->color) or „1" (color->black).

1. Approach

A 2D-integer-array is used to store the region number for each pixel. If the 4th case occurs, renumbering the whole integer array (due to two regions being equivalent), is very time consuming. Especially because this can happen more than once.

2. Approach

An equivalence map is used instead of the integer-array.

Region	0	1	2	3	4
Equivalent Region	0	1	1	0	4

Region equivalence Map

Problem: In case it is discovered that region „2" is not only equivalent to „1" but also to „3", this information would be overwritten. If renumbering is taking place immediately the processing time is raising again.

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Blob Coloring Algorithm cont.

3. Approach

Use a **`Vector`** of **`TreeSet`** objects.

Region	Equivalent Regions
0	
1	0
2	0,1
3	0
4	

Region equivalence Trees

Region	Equivalent regions	after flattening
0		
1	0	0
2	1	0
3	2	0
4	3	0
5	0	0
6	3, 5	0
7		
8	1	0
9	7	7
10	7	7
11	9	7
12	10	7

Region equivalence Table before and after flattening

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Blob Coloring Algorithm cont.

Determining which Blob to use

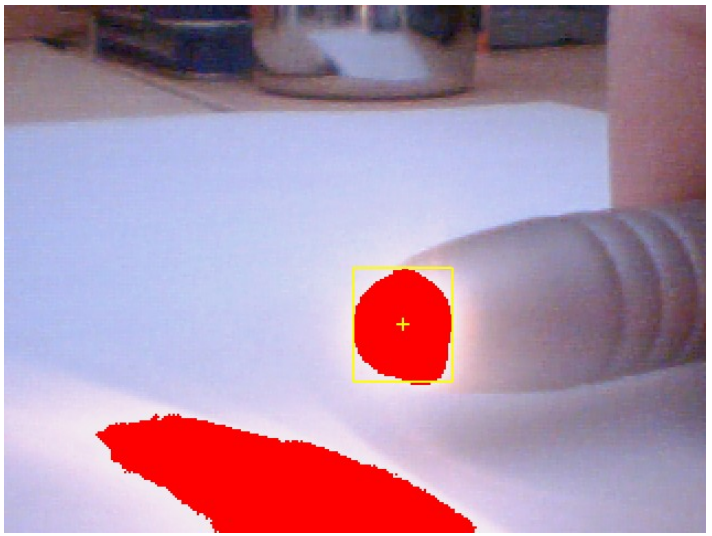
Problem: the biggest Blob is not always the one, which should be detected (i.e. large reflection from the surface).

Blob Criteria

- Absolute ratio between width and height of the blob bounding box.
- Minimum region size in pixels.

$$R_{wh} = |\frac{BoundingBox_{width}}{BoundingBox_{height}}|$$

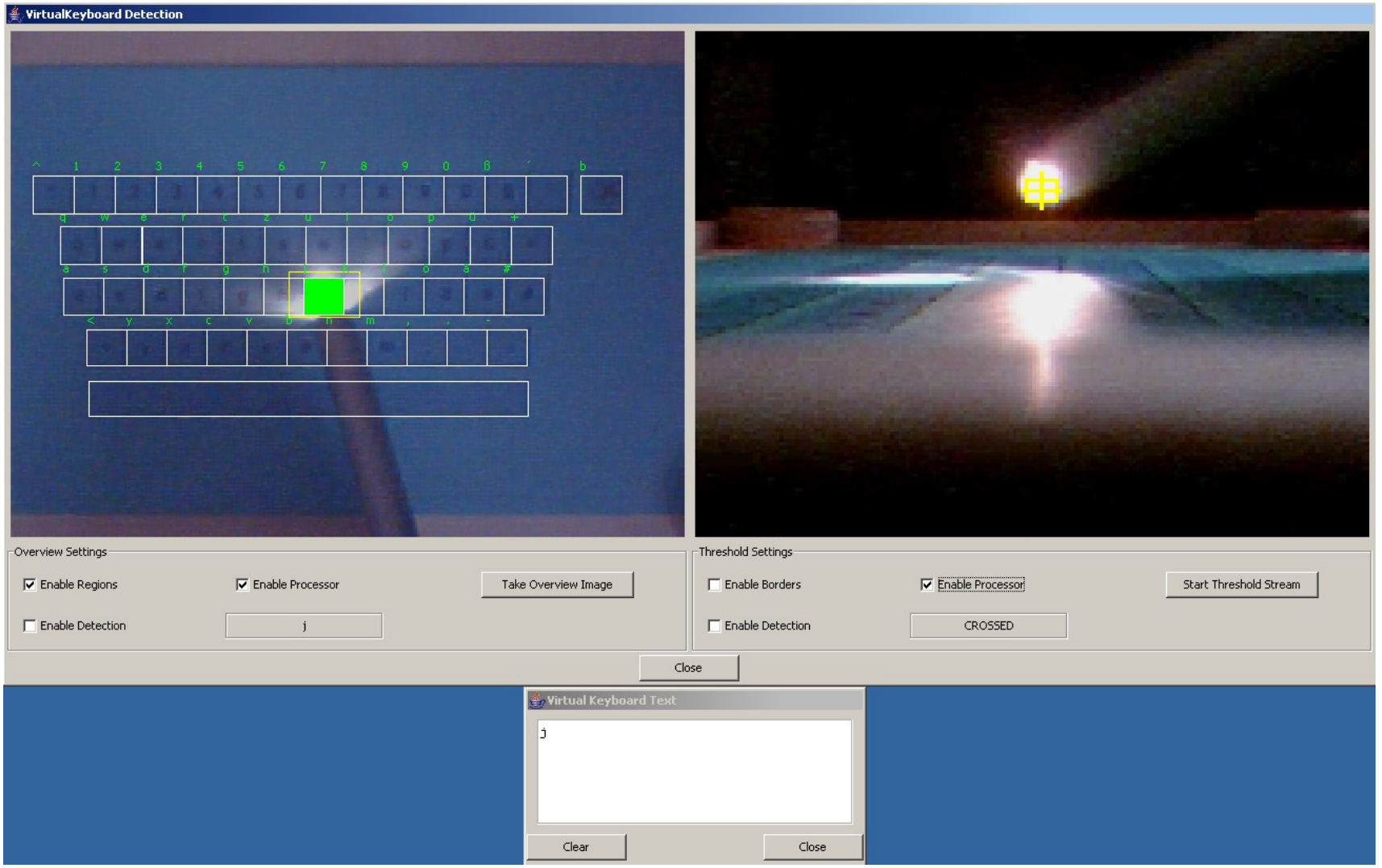
Bounding Box width-height Ratio



Detected Light Blob

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Tests and Results



Detection Test

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

fachhochschule münchen
munich university of applied sciences

Contents - Conclusion

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Conclusion

Results

All tests have been successfull if some constraints and problems are kept in mind.

Problems

- Switching cameras, taking the overview image and switching back takes about one to two seconds. This severly limits the input rate.
- Different light conditions have a big impact on the detection. Therefore the configuration needs to be checked frequently.
- Reflections on the keyboard surface still cause false-detections especially if the pen light combines with the reflection.

Future Work and Developments

- Updates might solve the problem with running both cameras at the same time resulting in a higher input rate.
- Other image processing algorithms should be considered for detection.
- Other non-image processing algorithms should be considered (i.e. *Celluon*).

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Bibliography

Papers

[1] C. Mehring, F. Kuester, S. Kunal Deep, M. Chen: „*KITTY: Keyboard Independent Touch Typing in VR*". IEEE Virtual Reality 2004.

Internet

[2] Canesta Inc., *Homepage*, April 2006, http://www.canesta.com/

[3] Canesta Inc., *Celluon*, April 2006, http://www.canesta.com/html/celluon.htm

[4] Stanford University Libraries & Academic Information Resources, *SiliconBase - The Mouse*, April 2006, http://www-sul.stanford.edu/siliconbase/wip/control.html

[5] Eleksen, *Homepage*, April 2006, http://www.eleksen.com/

[6] Eleksen, *Wireless fabric keyboard*, April 2006, http://www.eleksen.com/index.asp?page=products/wirelesskeyboard/keyboard_1.asp

[7] Hama GmbH & Co KG, *Homepage*, April 2006, http://www.hama.de/

[8] Hama GmbH & Co KG, *Bluetooth Freedom Mini Keyboard*, April 2006, http://www.hama.de/portal/articleId*126221/action*2563

[9] Garmin, *StreetPilot 2660*, April 2006, http://www.garmin.de/Produktbeschreibungen/StreetPilot2660.php

[10] Kiosk Mode, *Homepage*, April 2006, http://www.kiosk-mode.com/

[11] KITTY Tech, *Keyboard Independent Touch Typing Technology*, April 2006, http://www.kittytech.com/

[12] teltarif.de, *Siemens testet die virtuelle Tastatur*, 22nd March 2005, http://www.teltarif.de/arch/2005/kw12/s16621.html

Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)

Bibliography cont.

Internet

[13] Logitech, *QuickCam Express*, April 2006, http://www.logitech.com/index.cfm/products/details/DE/DE,CRID=2204,CONTENTID=5037

[14] Sun Microsystems, *Java Technology*, April 2006, http://java.sun.com/

[15] Sun Microsystems, *Mustang: Java SE 6*, April 2006, https://mustang.dev.java.net/

[16] Sun Microsystems, *Java Media Framework API (JMF)*, April 2006, http://java.sun.com/products/java-media/jmf/

[17] java.net Projects, *Java Media DirectShow (JMDS)*, April 2006, https://jmds.dev.java.net/

[18] Eclipse.org, *Homepage*, April 2006, http://www.eclipse.org

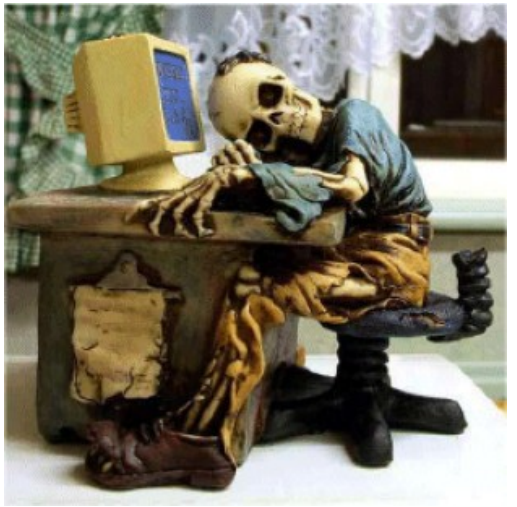
[19] David Bull, *Projects - Java - Colour Tracker*, http://www.uk-dave.com/projects/java/colourtracker.php

[20] Apache, *Log4j project*, April 2006, http://logging.apache.org/log4j/docs/index.html

Ende – End – Fin - Conclude

Thank you for your attention !

Questions ?



Thomas Wöllert (Dipl.-Inf.(FH)) (thomas@woellert.net)