

FabQR

Using QR Codes in the FabLab Workflow

Bachelor's thesis talk

July 15th, 2015

<http://hci.rwth-aachen.de/fabqr>

Supervisor:

Dipl.-Inform. René Bohne

Thesis advisor:

Prof. Dr. Jan Borchers

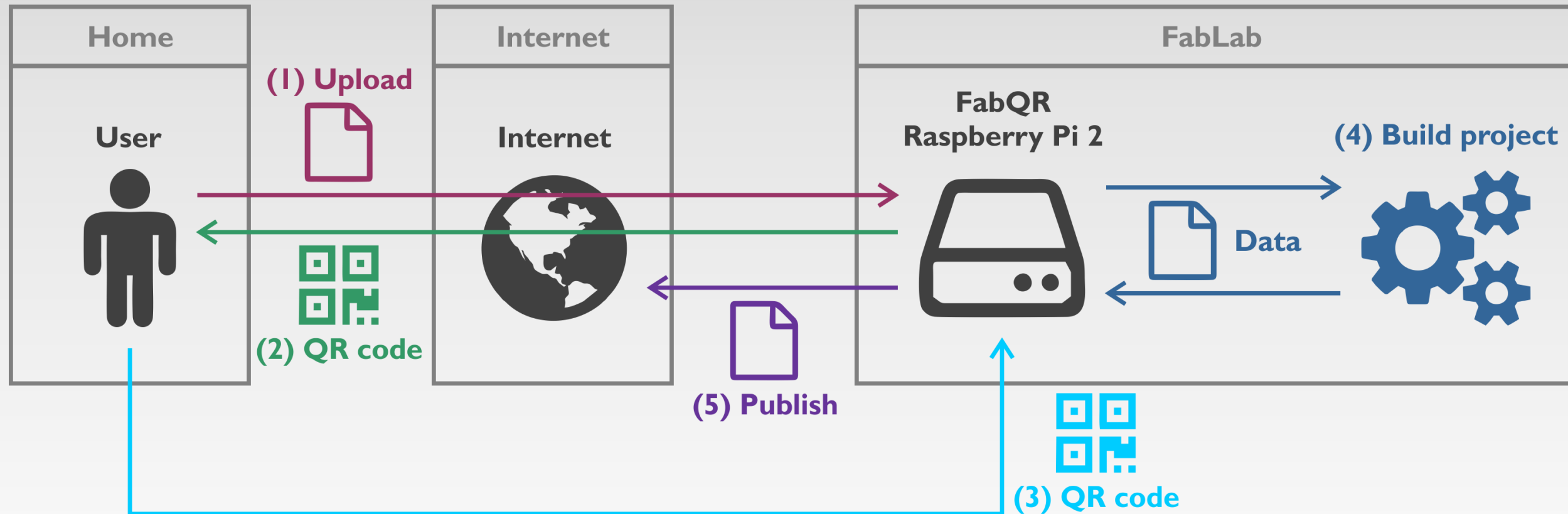
Second examiner:

Univ.-Prof. Dipl.-Ing. M.Arch Peter Russell

Contents

- ▶ Introduction
- ▶ Related Work
- ▶ Workflow
- ▶ Implementation
- ▶ Evaluation
- ▶ Future Work
- ▶ Summary

Introduction



Related Work

- ▶ Hardware: Raspberry Pi 2
- ▶ Graphics: openFrameworks
- ▶ Performance: OpenMAX
- ▶ QR codes: ZXing
- ▶ Laser cutting: VisiCut, VisiCam
- ▶ Web service: Apache, PHP

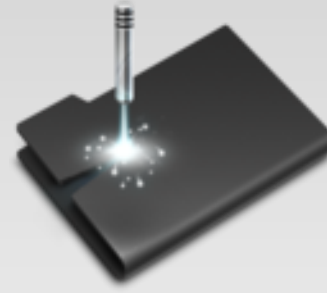
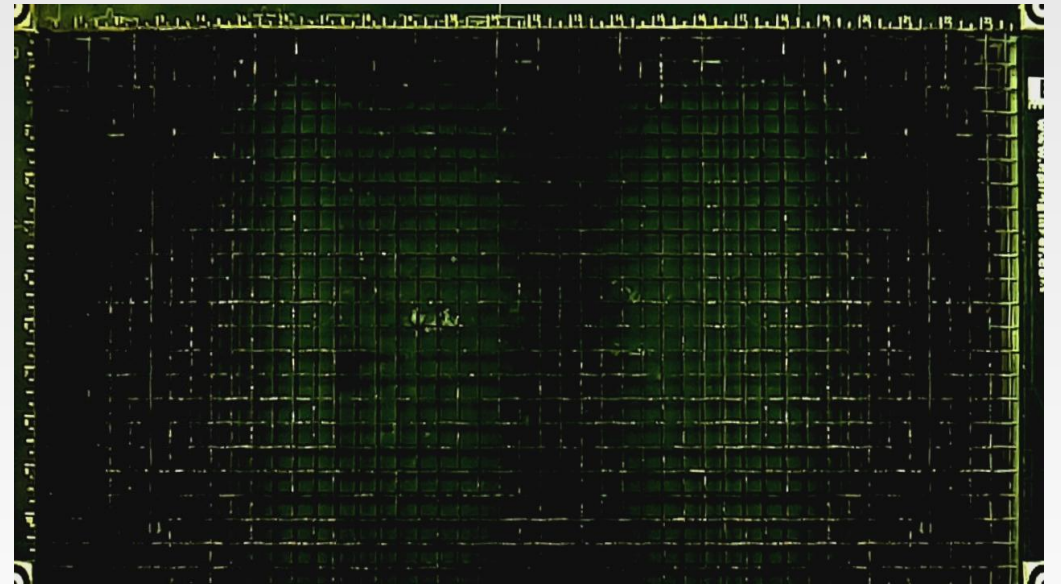


Image of the laser cutting area by VisiCut and VisiCam



Workflow – File Upload

File Upload

Select File

You can upload your project file to the FabLab here.
The file is temporarily stored for some weeks and it can only be accessed from the FabLab.
In the FabLab you can permanently publish your project after you have created it.
Maximum file size: 10 MB, Allowed file types: .plf,.svg

File Upload

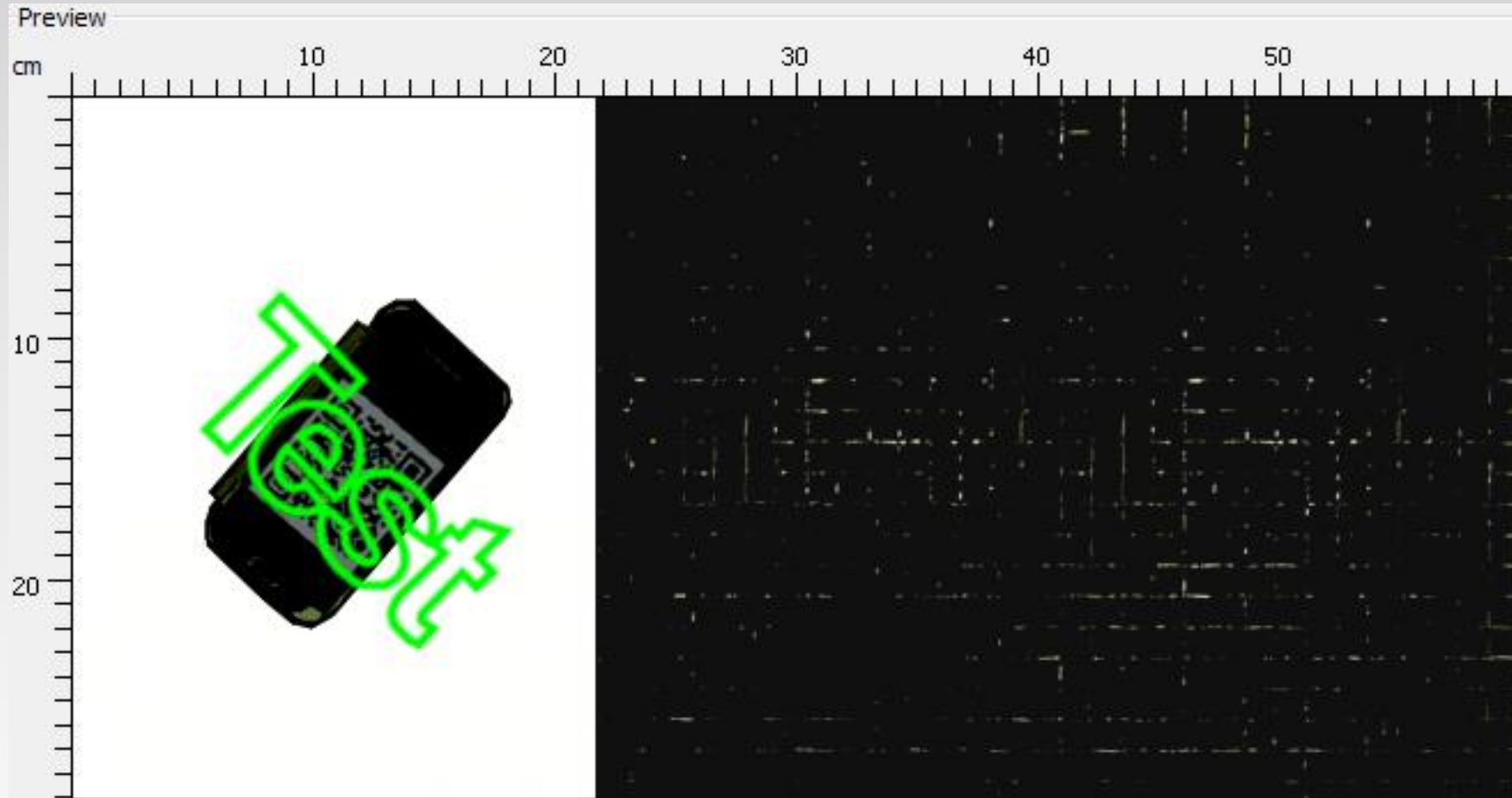


Your file was successfully uploaded.

Email QR Code

Print QR Code

Workflow – QR Code Detection



Interactive laser cutting with QR codes

Workflow – Project Documentation

Status: Waiting for user input

Name (optional):

E-Mail (optional):

Project name (min. 3 characters):

License (Creative Commons):

Tools:

☒ Laser cutter ☐ PCB / Soldering ☐ Arduino
☐ 3D printer ☐ CNC router ☐ Raspberry Pi


Description:

Take photo

Try again

Camera

☒ VisiCam ☐ Webcam



Close

Publish

Workflow – Website

Test project

Information



Name : Test Name
Email : email@example.org
License : CC0 1.0 (no restrictions)
Location : FabLab RWTH Aachen
Date : 6. July 2015, 15:34

Tools

- Laser cutter
 - Model : Epilog ZING
 - Material : Paper, 0.0 mm

Description

This is a test project. This text here will be published on the websites. The results of this project are shown in the photo.

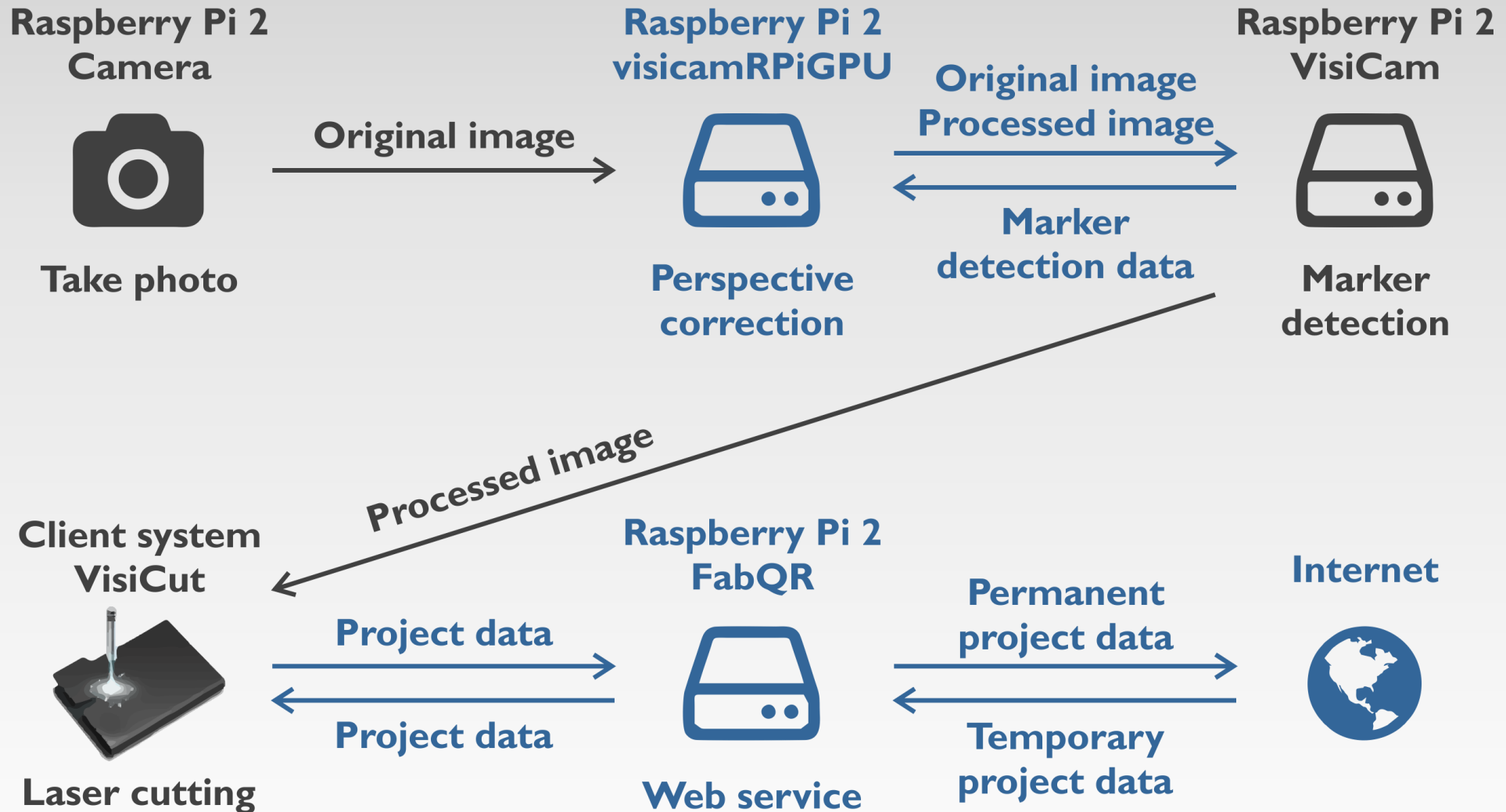


Download File

Email QR Code

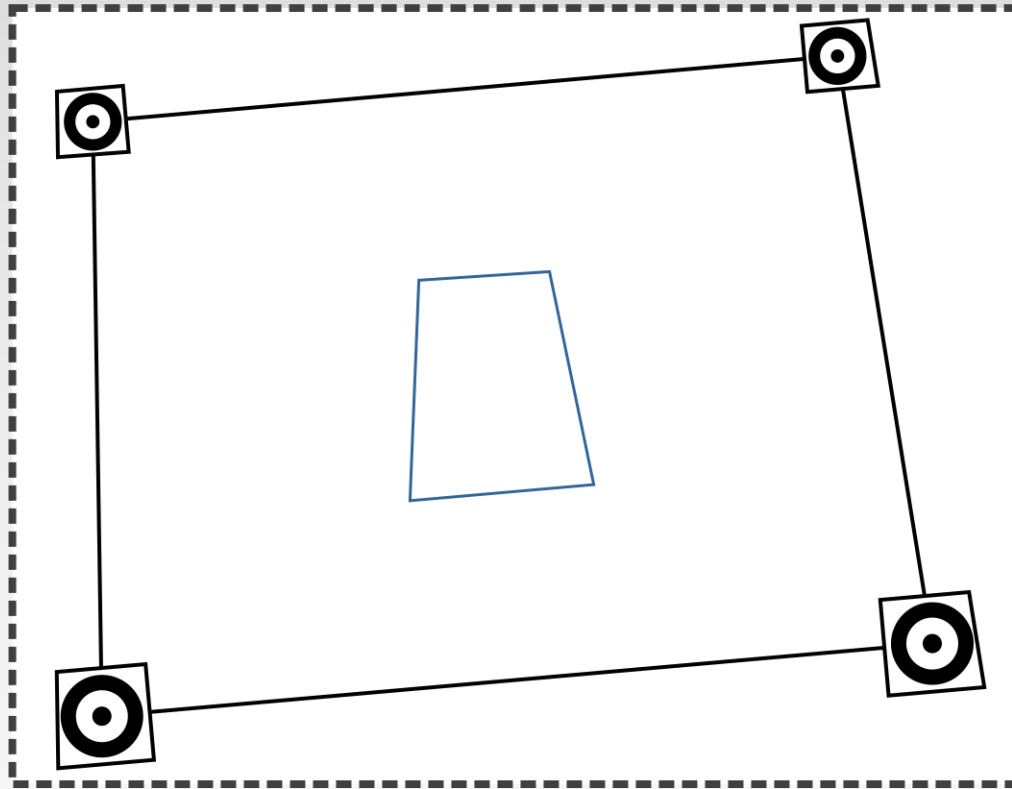
Print QR Code

Implementation – System Overview

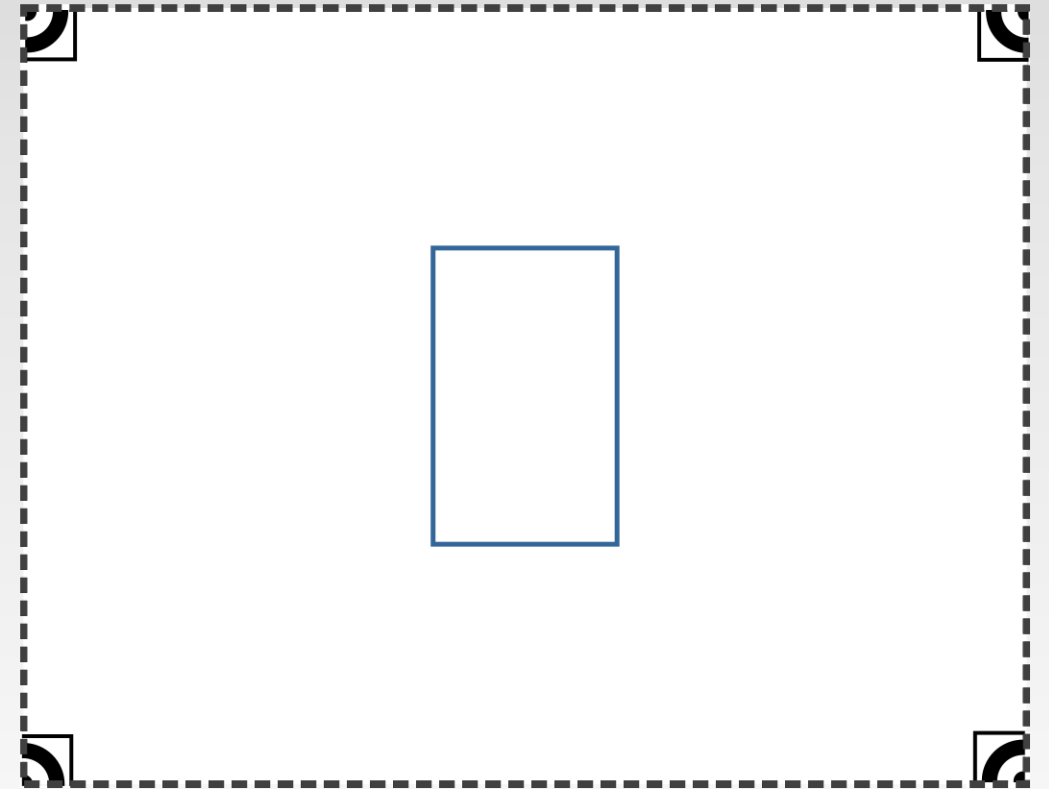


Implementation – Perspective Correction

Original image



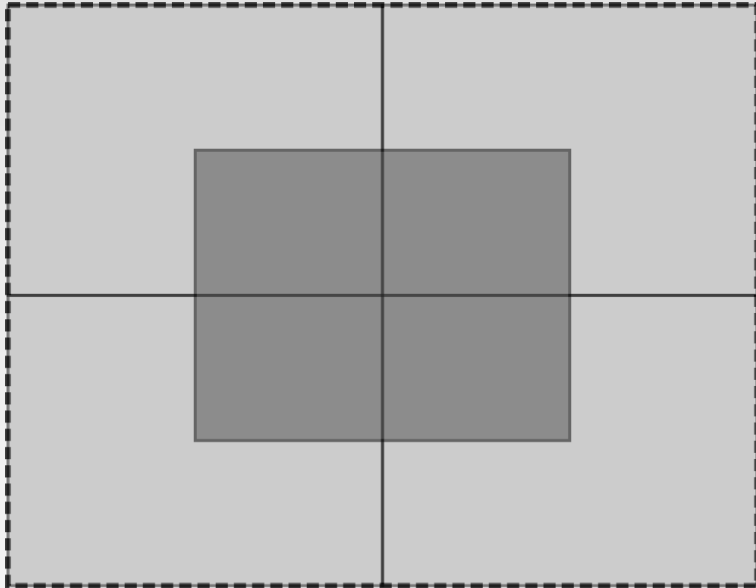
Perspective corrected image



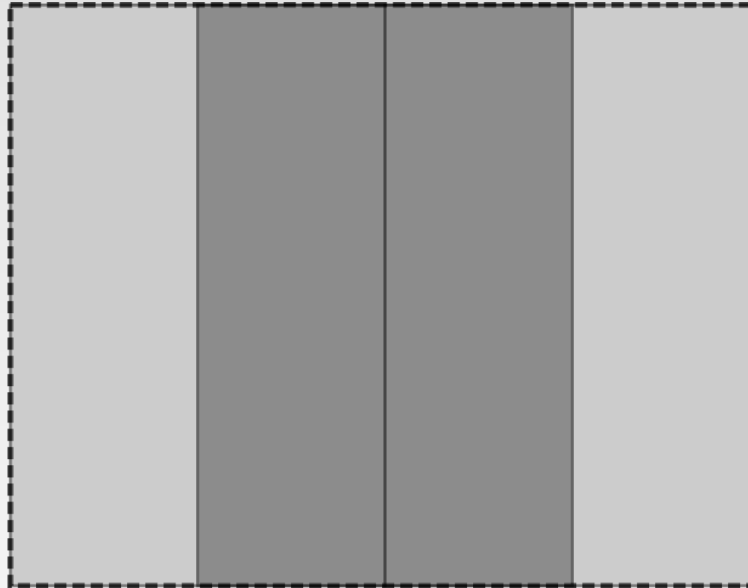
Performance improvements: 100 ms instead of 8000 ms

Implementation – QR Code Detection

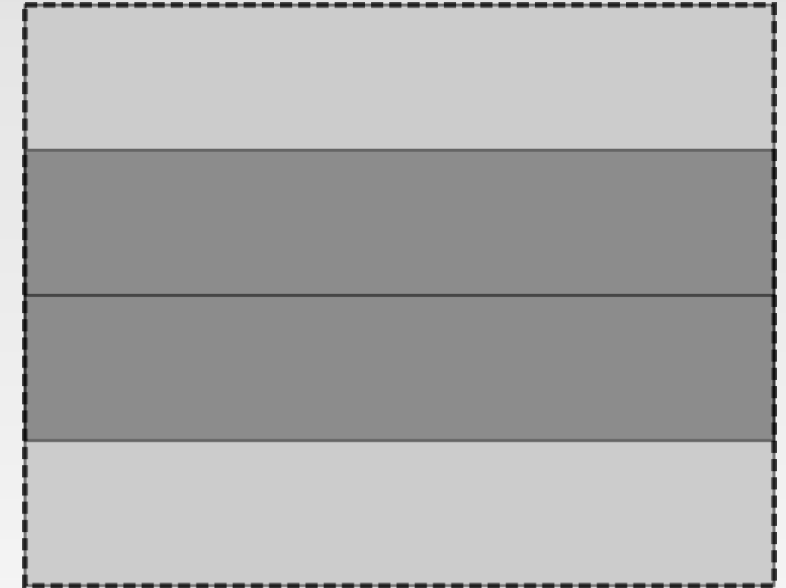
Section mode
Quadrants and Center



Slice mode
3 vertical slices



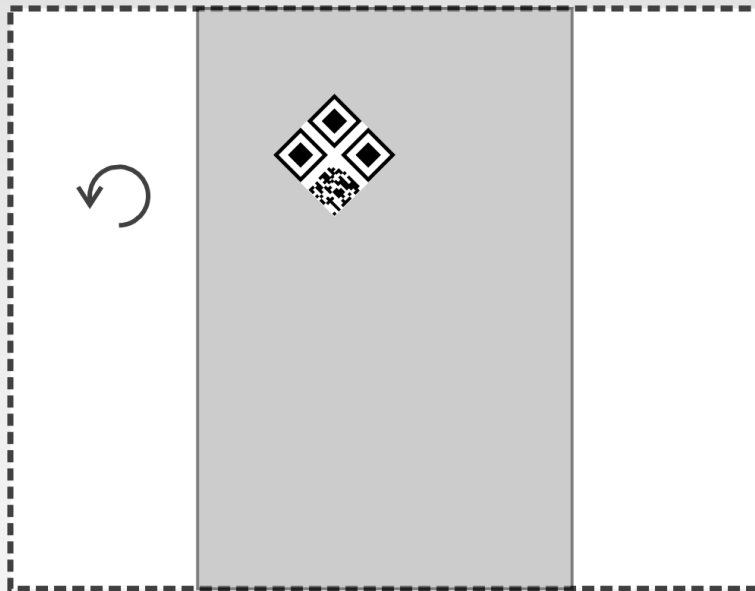
Slice mode
3 horizontal slices



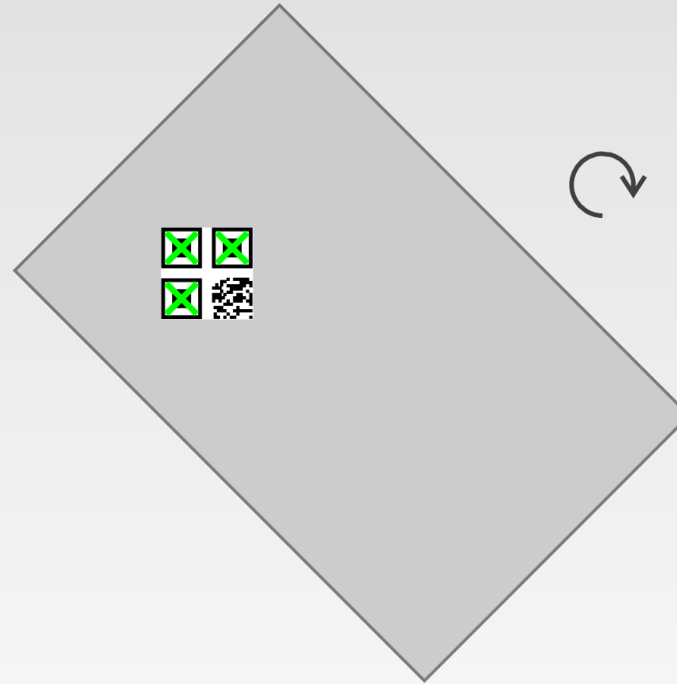
Advanced algorithm: Detection of multiple QR codes

Implementation – QR Code Detection

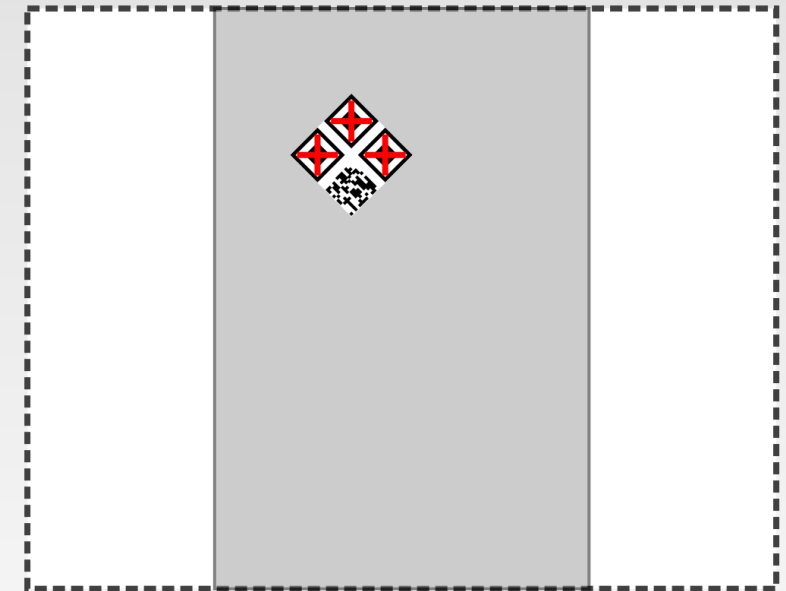
Undetected QR code



Rotated image part



Detected QR code with coordinates



Advanced algorithm: Detection of rotated QR codes

Evaluation

- ▶ System Usability Scale
 - ▶ Questionnaire: 10 items
 - ▶ Quantifies usability of a system
 - ▶ Range: 0 to 100
- ▶ User study: 12 participants, P4 is outlier
- ▶ Average result: 70.83
- ▶ Average result without P4: 74.77

Participants and results of the user study

P1	87.5	P7	80.0
P2	62.5	P8	82.5
P3	80.0	P9	80.0
P4	27.5	P10	72.5
P5	75.0	P11	75.0
P6	72.5	P12	55.0

Future Work

- ▶ Further performance improvements
- ▶ Additional website features
- ▶ Stand-alone FabQR client
- ▶ File sharing network

Summary

- ▶ FabLab workflow can be improved
- ▶ Evaluation: FabQR has potential to improve workflow

***Physical file sharing with QR codes
improves the FabLab workflow.***

<http://hci.rwth-aachen.de/fabqr>