Semester Thesis: "Object Recognition with Mobile Phones"

Marcel Germann

February 16, 2006
Supervisor: Till Quack

1. Introduction
2. Approach 1
3. Approach 2
4. Results
5. Conclusions

Mobile Phone Technologies

Mobile Phone Technologies

- SMS, MMS

Mobile Phone Technologies

- SMS, MMS
- infrared, bluetooth

Mobile Phone Technologies

- SMS, MMS
- infrared, bluetooth
- Cameras

Combine OR with Mobile Phones

Combine OR with Mobile Phones

- Use as input a picture taken with mobile phone camera

Combine OR with Mobile Phones

- Use as input a picture taken with mobile phone camera
- Send it to a server which applies OR algorithm

Combine OR with Mobile Phones

- Use as input a picture taken with mobile phone camera
- Send it to a server which applies OR algorithm
- Server sends information based on the result back to the users mobile phone

Applications

Applications

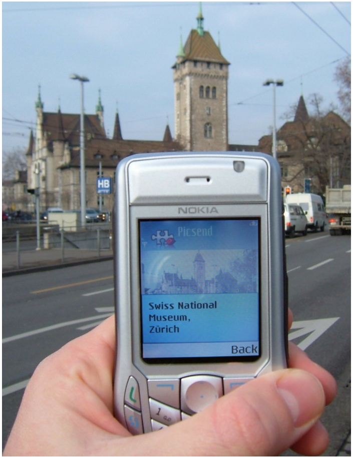
- Advertising campains

Applications

- Advertising campains
- Museum guide

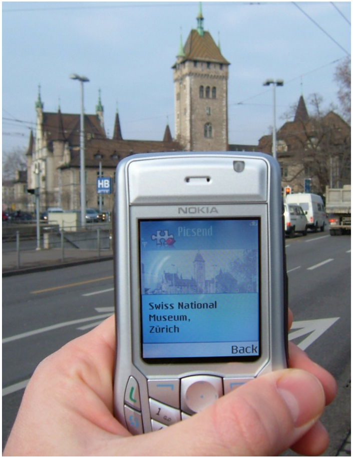
Applications

- Advertising campains
- Museum guide
- City guide (no barcode needed)



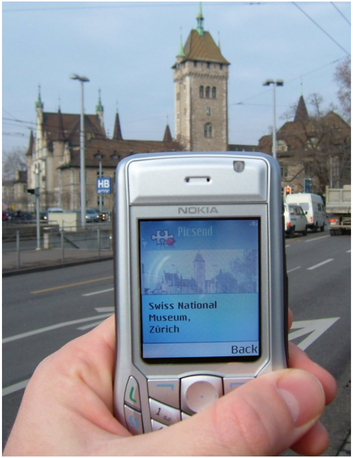
Applications

- Advertising campains
- Museum guide
- City guide (no barcode needed)
- Product information (books, etc.)



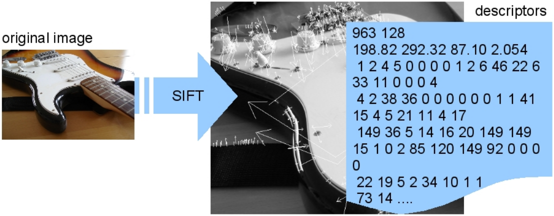
Applications

- Advertising campains
- Museum guide
- City guide (no barcode needed)
- Product information (books, etc.)
- Help for the blind



SIFT

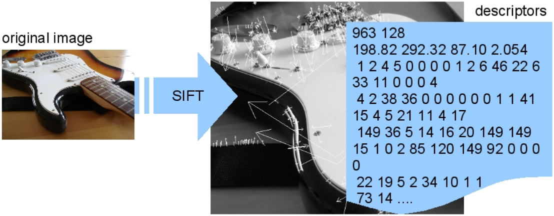
Scale Invariant Feature Transform (SIFT) by David G. Lowe



```
963 128
198.82 292.32 87.10 2.054
 1 2 4 5 0 0 0 0 1 2 6 46 22 6
33 11 0 0 0 4
 4 2 38 36 0 0 0 0 0 0 1 1 41
15 4 5 21 11 4 17
 149 36 5 14 16 20 149 149
15 1 0 2 85 120 149 92 0 0 0
0
 22 19 5 2 34 10 1 1
 73 14 ....
```

SIFT

Scale Invariant Feature Transform (SIFT) by David G. Lowe

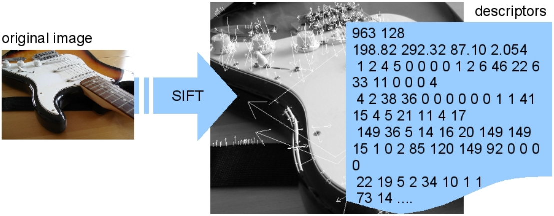


```
963 128
198.82 292.32 87.10 2.054
 1 2 4 5 0 0 0 0 1 2 6 46 22 6
33 11 0 0 0 4
 4 2 38 36 0 0 0 0 0 0 1 1 41
15 4 5 21 11 4 17
 149 36 5 14 16 20 149 149
15 1 0 2 85 120 149 92 0 0 0
0
 22 19 5 2 34 10 1 1
 73 14 ....
```

Features:

SIFT

Scale Invariant Feature Transform (SIFT) by David G. Lowe



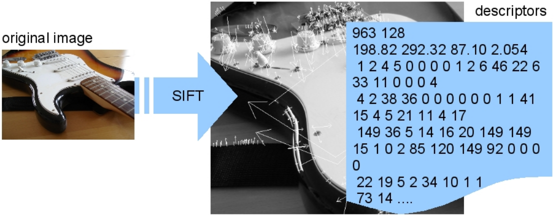
```
963 128
198.82 292.32 87.10 2.054
 1 2 4 5 0 0 0 0 1 2 6 46 22 6
33 11 0 0 0 4
 4 2 38 36 0 0 0 0 0 0 1 1 41
15 4 5 21 11 4 17
 149 36 5 14 16 20 149 149
15 1 0 2 85 120 149 92 0 0 0
0
 22 19 5 2 34 10 1 1
 73 14 ....
```

Features:

- Scale invariant

SIFT

Scale Invariant Feature Transform (SIFT) by David G. Lowe



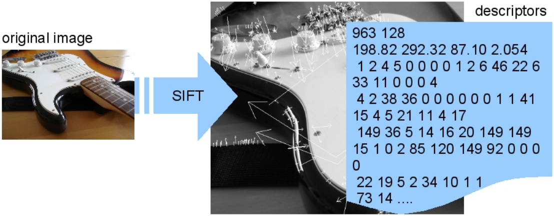
```
963 128
198.82 292.32 87.10 2.054
 1 2 4 5 0 0 0 0 1 2 6 46 22 6
33 11 0 0 0 4
 4 2 38 36 0 0 0 0 0 0 1 1 41
15 4 5 21 11 4 17
 149 36 5 14 16 20 149 149
15 1 0 2 85 120 149 92 0 0 0
0
 22 19 5 2 34 10 1 1
 73 14 ....
```

Features:

- Scale invariant
- Rotation invariant

SIFT

Scale Invariant Feature Transform (SIFT) by David G. Lowe



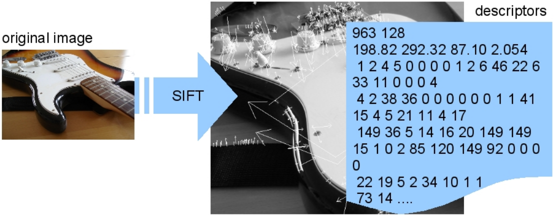
```
963 128
198.82 292.32 87.10 2.054
 1 2 4 5 0 0 0 0 1 2 6 46 22 6
33 11 0 0 0 4
 4 2 38 36 0 0 0 0 0 0 1 1 41
15 4 5 21 11 4 17
 149 36 5 14 16 20 149 149
15 1 0 2 85 120 149 92 0 0 0
0
 22 19 5 2 34 10 1 1
 73 14 ....
```

Features:

- Scale invariant
- Rotation invariant
- Robust against noise and change in illumination

SIFT

Scale Invariant Feature Transform (SIFT) by David G. Lowe



```
963 128
198.82 292.32 87.10 2.054
 1 2 4 5 0 0 0 0 1 2 6 46 22 6
33 11 0 0 0 4
 4 2 38 36 0 0 0 0 0 0 1 1 41
15 4 5 21 11 4 17
 149 36 5 14 16 20 149 149
15 1 0 2 85 120 149 92 0 0 0
0
 22 19 5 2 34 10 1 1
 73 14 ....
```

Features:

- Scale invariant
- Rotation invariant
- Robust against noise and change in illumination
- Also small changes in 3D viewpoint are possible

Two Approaches

Two Approaches

- Approach 1: server side calculation

Two Approaches

- Approach 1: server side calculation
- Approach 2: calculation fully on mobile phone

Approach 1: Server Side Calculation

Approach 1: Server Side Calculation

- Do the expensive algorithms on a server

Approach 1: Server Side Calculation

- Do the expensive algorithms on a server
- Take the picture with mobile phone and send it to the server

Approach 1: Server Side Calculation

- Do the expensive algorithms on a server
- Take the picture with mobile phone and send it to the server
- No need to install something on the mobile phone

Approach 1: Server Side Calculation

- Do the expensive algorithms on a server
- Take the picture with mobile phone and send it to the server
- No need to install something on the mobile phone
- Large database possible

Communication with Server

Transmission of a picture from the mobile phone to the server.
Possibilities:

Communication with Server

Transmission of a picture from the mobile phone to the server.
Possibilities:

- **bluetooth, infrared**: no real mobility

Communication with Server

Transmission of a picture from the mobile phone to the server.
Possibilities:

- **bluetooth, infrared**: no real mobility
- **internet**: bad usability and rarely supported

Communication with Server

Transmission of a picture from the mobile phone to the server.
Possibilities:

- **bluetooth, infrared**: no real mobility
- **internet**: bad usability and rarely supported
- **MMS**: standard on most mobile phones

Approach 1: Server Side Calculation

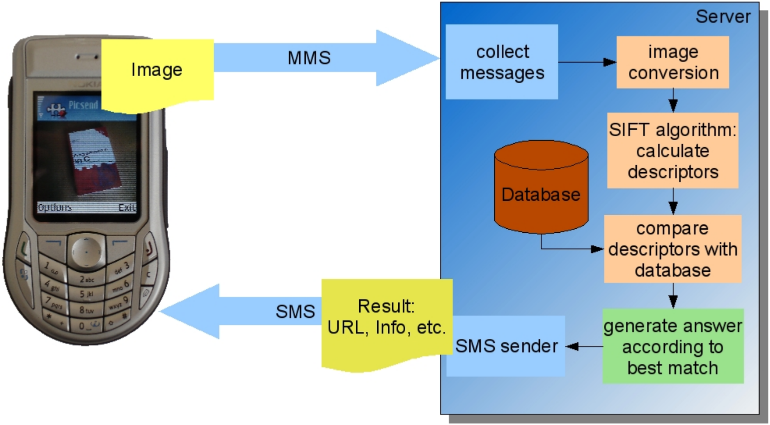


Image
MMS
Server
collect messages
image conversion
SIFT algorithm: calculate descriptors
Database
compare descriptors with database
generate answer according to best match
SMS sender
Result: URL, Info, etc.
SMS
Picsend
Options
Exit

Symbian Application

Symbian Application

- Small application that runs on the mobile phone

Symbian Application

- Small application that runs on the mobile phone
- Functions: take picture, send picture

Symbian Application

- Small application that runs on the mobile phone
- Functions: take picture, send picture
- Better compatibility

Implementation

Implementation

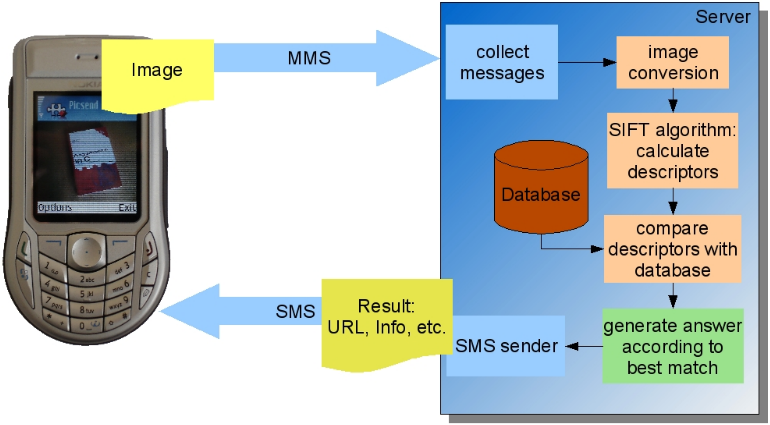


Image
MMS
Picsend
Options
Exit
Server
collect messages
image conversion
SIFT algorithm: calculate descriptors
Database
compare descriptors with database
generate answer according to best match
SMS sender
SMS
Result: URL, Info, etc.

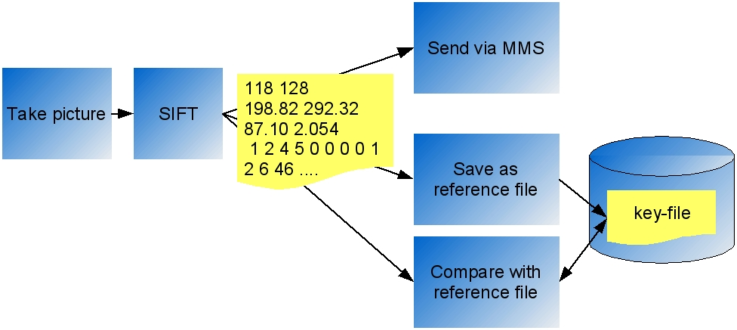
Approach 2: Calculation Fully on Mobile Phone

Approach 2: Calculation Fully on Mobile Phone

One application that runs completely independent on a mobile phone.

Approach 2: Calculation Fully on Mobile Phone

One application that runs completely independent on a mobile phone.



Implementation

- Implementation on Symbian v8.0

Implementation

- Implementation on Symbian v8.0
- Symbian is 'kind of' C++

Implementation

- Implementation on Symbian v8.0
- Symbian is 'kind of' C++
- Problems with Symbian:

Implementation

- Implementation on Symbian v8.0
- Symbian is 'kind of' C++
- Problems with Symbian:
    - Error handling

Implementation

- Implementation on Symbian v8.0
- Symbian is 'kind of' C++
- Problems with Symbian:
  - Error handling
  - No global writable variables

Implementation

- Implementation on Symbian v8.0
- Symbian is 'kind of' C++
- Problems with Symbian:
  - Error handling
  - No global writable variables
  - Very very small stack

Implementation

- Implementation on Symbian v8.0
- Symbian is 'kind of' C++
- Problems with Symbian:
    - Error handling
    - No global writable variables
    - Very very small stack
    - Heap is limited at 5 MB. This is e.g. too small to hold more than four 640*480 pictures with float pixels!

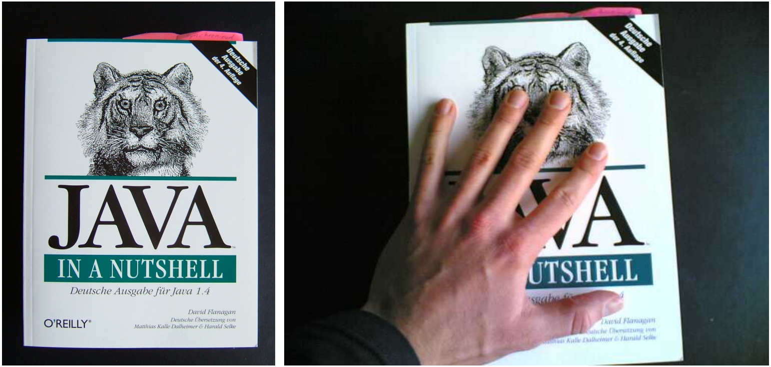
Implementation

- Implementation on Symbian v8.0
- Symbian is 'kind of' C++
- Problems with Symbian:
  - Error handling
  - No global writable variables
  - Very very small stack
  - Heap is limited at 5 MB. This is e.g. too small to hold more than four 640*480 pictures with float pixels!
- Application calculation with different image sizes: 320*240 (slower) and 160*120 (faster)

Results - Approach 1

Results - Approach 1

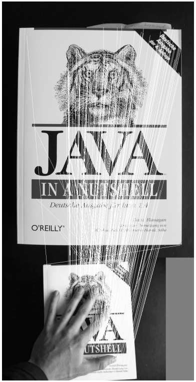
Specific example:


Deutsche Ausgabe der 4. Auflage
JAVA
IN A NUTSHELL
Deutsche Ausgabe für Java 1.4
David Flanagan
Deutsche Übersetzung von
Matthias Kalle Dalheimer & Harald Selke
O'REILLY

Left: Original picture gives 3233 keypoints. Right: Picture received by MMS gives 853 keypoints.

Results - Approach 1

Server side matching:



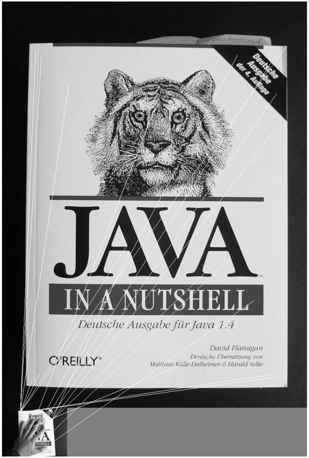
⇒ 268 matches

Results - Approach 2

Keypoints calculated on the mobile:
Fast calculation (image size 160*120) takes ≈ 13 sec and gives 35 keypoints.

Results - Approach 2

Keypoints calculated on the mobile:
Fast calculation (image size 160*120) takes ≈ 13 sec and gives 35 keypoints.
Result: ⇒ 25 matches

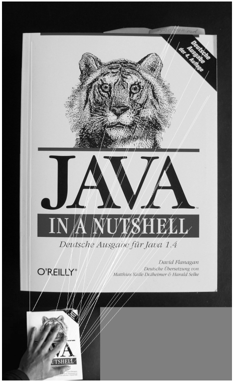


Results - Approach 2

Keypoints calculated on the mobile:
Slow calculation (image size 320*240) takes ≈ 50 sec and gives 61 keypoints.

Results - Approach 2

Keypoints calculated on the mobile:
Slow calculation (image size 320*240) takes ≈ 50 sec and gives 61 keypoints.
Result: ⇒ 27 matches



Results

Demo ...

Conclusions

Conclusions

- Framework 'mobileor' is a basis for mobile object recognition.

Conclusions

- Framework 'mobileor' is a basis for mobile object recognition.
- Results are promising - even those with keypoints calculated on the mobile phone.

Conclusions

- Framework 'mobileor' is a basis for mobile object recognition.
- Results are promising - even those with keypoints calculated on the mobile phone.
- Server side approach can be used to e.g. build up a city guide.

Future Work

Future Work

- Solutions for a large database

Future Work

- Solutions for a large database
- Specific application (e.g. city guide)

Future Work

- Solutions for a large database
- Specific application (e.g. city guide)
- Improve the SIFT implementation on the mobile phone

Future Work

- Solutions for a large database
- Specific application (e.g. city guide)
- Improve the SIFT implementation on the mobile phone
- SURF instead of SIFT

End

Thank you!