Implementation and Evaluation of a mobile VNC Client with Multitouch Transfer and Gesture Recognition System using the Android SDK

— Final presentation of bachelor's thesis —

Richard Haus
February 9, 2015
Computer Science Dept. 5 (Pattern Recognition)
Friedrich-Alexander University Erlangen-Nuremberg
Advisors: Dr. Ing- Stefan Steidl, Mario Amrehn M. Sc.

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Table of Contents

Motivation

Need for a Multitouch VNC Client

- **In General:** More intuitive handling during remote control
- **Economy:** Controlling machine tools from afar

Fig.: Telematic Control Unit (TCU)

Need for a Multitouch VNC Client

- **In General:** More intuitive handling during remote control
- **Economy:** Controlling machine tools from afar
- **Private:** Better handling of multimendia applications (photo viewer)

Fig.: Windows Photo Viewer

Need for a Gesture Recognition AddOn

- improved user interface
  ⇒ no menu crawling for special action
  e.g. sending special key combinations

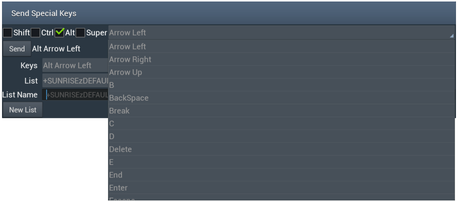
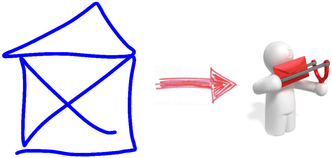


Fig.: open menu tree of such a mobile VNC Client

Need for a Gesture Recognition AddOn

- improved user interface
  $\Rightarrow$ no menu crawling for special action
  e.g. sending special key combinations

- huge spectrum of realizable functions without a seperate GUI
  e.g. gesture "Santas House" causes mail
  with certain content to predefined person



Methods

Client Implementation of Multitouch Transfer

- Internalize existing VNC functionallity - the RFB protocol
  1. **Handshake**
     arrange connection parameters
     authentication, encoding, color depth, screen resolution
  2. ***Server*: Framebuffer updates**
     send only regions of change
     encoded with agreed encoding per default, uncompressed at tiny areas
  3. ***Client*: controlling**
     send instructions
     mouse moving, keyboard input, update connection parameters if required

Client Implementation of Multitouch Transfer

- Internalize existing VNC functionallity - the RFB protocol
- Extend the protocol with the modified GII

In advance
define datastrcture for Multitouch input
array of single touch points:

⋮

x-pos	y-pos	pressed flag	primary flag	timestamp

⋮

During handshake
arrage GII using with the server
by adding extra encoding type

Client Implementation of Multitouch Transfer

- Internalize existing VNC functionallity - the RFB protocol
- Extend the protocol with the modified GII
- Wrap Multitouch input on device into protocol
    1. Pick up single touch points in callback function
    2. Convert the position to server screen space (scale & shift)
    3. Assign global unique finger ID's
    4. Treat special cases
    5. Pack points in datastructure and send

Gesture Detection AddOn for Multitouch Client

So far:

- 👍 Multitouch transfer
  comfortable handling of Multitouch applications on server

- 👎 No possibility for shifting or scaling the image section
- 👎 Sending key combinations as difficult as before

Gesture Detection AddOn for Multitouch Client

So far:

- 👍 Multitouch transfer
  comfortable handling of Multitouch applications on server
- 👎 No possibility for shifting or scaling the image section
- 👎 Sending key combinations as difficult as before

General Problem: **Issue of membership** of display input
No chance to decide, whether input belongs to local control or to server

Issue of Membership

Flagfield := rectangular area on corner of display

- Has two states like binary flag: **SET**$_{\text{(pressed)}}$ or **UNSET**$_{\text{(not pressed)}}$
- If **SET** ("activated") ⇒ touch input gets redirected for special purpose

Local Control Flagfield

Remote Control Flagfield

Issue of Membership

Activation via **double click** into its area

☞ Same difficulty as original problem of membership
(double click could be intended for controlling the server)

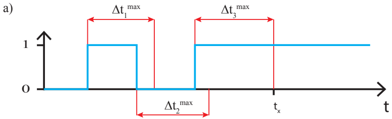
Issue of Membership

Activation via **double click** into its area

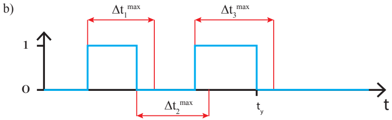
☞ Same difficulty as original problem of membership (double click could be intended for controlling the server)

Solution with time limits and input buffering

signal = 1 ⇔ finger on flagfield



succesfull flagfield activation



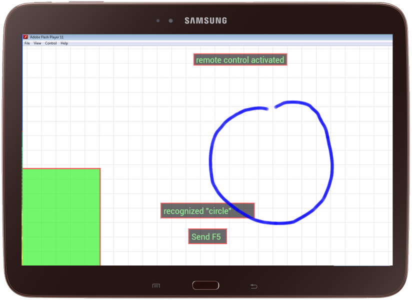
succesfull transferred double click

Remote Control

Effect server reaction by sending keyboard shortcuts
► Choose reaction by drawing certain gesture

If activated by *Remote Control Flagfield* all further touch inputs are redirected into *Gesture Recognition System*

- classifier delivers class label
- mapping from label to reaction
- send accordant combination of (meta) keys



Remote Control - Mapping from class laben to action

gesture	shortcut to send	function
”circle”	*F5*	update or reload
”triangle”	*Ctrl+Alt+Del*	open taskmanager
”audi”	*Super + D*	swap to desktop
”dia to upper left”	*Escape*	escape
”zoom in”	*Ctrl + ’+’*	zoom in screen
”zoom out”	*Ctrl + ’-’*	zoom out screen
”double circle”	*Ctrl + 0*	reset zoom to 100%
”pan left” $\leftarrow$	*Arrow left*	move left
”pan right” $\rightarrow$	*Arrow right*	move right
”pan up” $\uparrow$	*Arrow up*	move upwards
”pan down” $\downarrow$	*Arrow down*	move downwards
”hook up”	*Page up*	scroll page up
”hook down”	*Page down*	scroll page down
”two finger circle”	$Alt + Tab_{(without\ release)}$	choose other application
”double hook down”	$Alt + Tab_{(release)}$	jump to choosen application

Local Control

Purpose: Scale or Shift the image section

If activated by *Local Control Flagfield* two functions are availible

- 1 finger on the display: Moving image section with this finger
- 2 fingers on the display: Scale image section by distance of the fingers

  ($>$ 2 fingers on the display: no reaction)

Local Control

Purpose: Scale or Shift the image section

If activated by *Local Control Flagfield* two functions are availible

- 1 finger on the display: Moving image section with this finger
- 2 fingers on the display: Scale image section by distance of the fingers

($>$ 2 fingers on the display: no reaction)

***Local Control* cannot be integrated into *Gesture Recognition System*!**

- Loss of direct feedback
- Need for unlimited gestures for each panning direction and scaling size

Need for a well Trained Classifier for Gesture Recognition

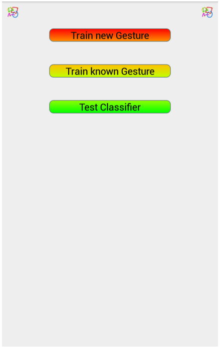
VNC App is not the right environment for training and defining new gestures

$\Rightarrow$ seperate **GesDe-Application**

Need for a well Trained Classifier for Gesture Recognition

VNC App is not the right environment for training and defining new gestures

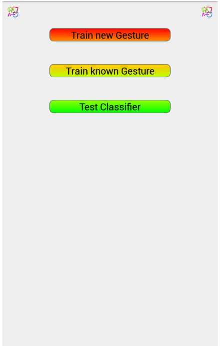
⇒ seperate **GesDe-Application**

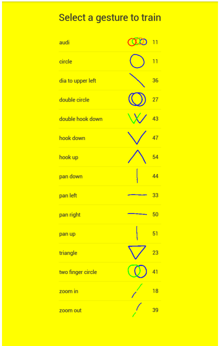


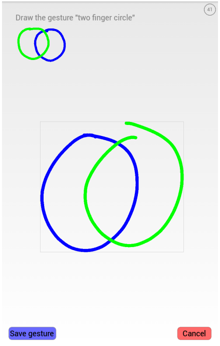
Need for a well Trained Classifier for Gesture Recognition

VNC App is not the right environment for training and defining new gestures

⇒ seperate **GesDe-Application**







Implementation of the Pipeline of Gesture Recognition

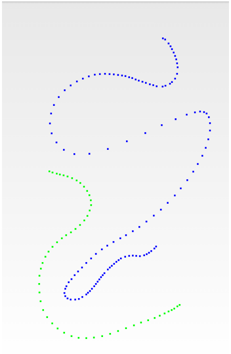
Classical setup of a gesture recognition system:

1. **Recording**
2. **Preprocessing**
3. **Feature extraction**
4. **Training**
5. **Classification**

Implementation of the Pipeline of Gesture Recognition

Recording & Proprocessing (merged)

Store all single touch points of one finger in a separate list



N lists $\mathbf{f}_k$ each with n_k points $P_{k,i} \in \mathbb{N}^3 \begin{pmatrix} P_{k,i(x)} \\ P_{k,i(y)} \\ \tau_{k,i}[msec] \end{pmatrix}$

$\mathbf{f}_0$:= list of finger_0	$\{P_{0,0},$	$P_{0,1}$,	$\ldots,$	$P_{0,n_0-1}\}$
$\mathbf{f}_1$:= list of finger_1	$\{P_{1,0},$	$P_{1,1},$	$\ldots,$	$P_{1,n_1-1}\}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\mathbf{f}_k$:= list of finger_k	$\{P_{k,0},$	$P_{k,1},$	$\ldots,$	$P_{k,n_k-1}\}$

additional:
list of distances of synchronous occuring single points

Implementation of the Pipeline of Gesture Recognition

Feature extraction

Create feature vector with 46 entries (incl. class label)

Implementation of the Pipeline of Gesture Recognition

Feature extraction *Number of corners*

first approach

search for small curvature angles

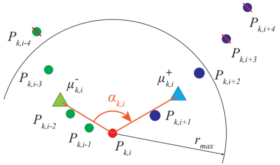


Fig.: estimated curvature angle at $P_{k,i}$

further tests for local minima of angles

Implementation of the Pipeline of Gesture Recognition

Feature extraction *Number of corners*

first approach

search for small curvature angles

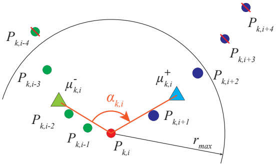


Fig.: estimated curvature angle at $P_{k,i}$

further tests for local minima of angles

second approach

search for strong bending

- normalize point list to arc length
- approximate first derivative

$$P'_{k,i} = P_{k,i+1} - P_{k,i-1} \quad (1)$$

- compute second derivative

$$P''_{k,i} = P'_{k,i+1} - P'_{k,i-1} \quad (2)$$

- look for local maxima of $||P''_{k,i}||_2$

Implementation of the Pipeline of Gesture Recognition

Feature extraction *Closeness of finger traces*

For differentiation between opened and closed pattern

$$f_6 := \frac{1}{BB_{dia}} \cdot \left\| P_{0,n_0-1} - P_{0,0} \right\|_2$$ (primary finger)

$$f_{22} := \frac{1}{BB_{dia}} \cdot \frac{1}{F} \sum_{k=0}^{F-1} \left\| P_{k,n_k-1} - P_{k,0} \right\|_2$$ (average over all finger traces)

Normalize with lenth of diagonal of bounding box to abstract from the size of the gesture

$$BB_{dia} = \left\| \begin{pmatrix} BB_{Width} \\ BB_{Height} \end{pmatrix} \right\|_2 = \left\| \begin{pmatrix} \max\limits_{\substack{0 \leq k < F \\ 0 \leq i < n_k}} \left(P_{k,i(x)} \right) - \min\limits_{\substack{0 \leq k < F \\ 0 \leq i < n_k}} \left(P_{k,i(x)} \right) \\ \max\limits_{\substack{0 \leq k < F \\ 0 \leq i < n_k}} \left(P_{k,i(y)} \right) - \min\limits_{\substack{0 \leq k < F \\ 0 \leq i < n_k}} \left(P_{k,i(y)} \right) \end{pmatrix} \right\|_2$$

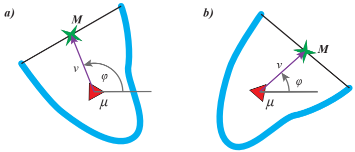
Implementation of the Pipeline of Gesture Recognition

Feature extraction *Opening direction of gesture*

Inspect vector v
from focus μ
to midpoint of endpoints M

$$\mu = \frac{1}{F} \sum_{k=0}^{F-1} \frac{1}{n_k} \sum_{i=0}^{n_k-1} P_{i,k}$$

$$M = \frac{1}{2} \cdot \left(P_{0,0} + P_{0,n_0-1}\right)$$



$$f_7 := \frac{1}{BB_{dia}} \cdot \|\mathbf{v}\|_2, \text{ mit } \mathbf{v} = M - \mu$$

$$f_8 := \alpha = \sphericalangle M\mu\left(\mu + \binom{1}{0}\right)$$

Implementation of the Pipeline of Gesture Recognition

Feature extraction *Length of touch traces*

$$len_k = \sum_{i=1}^{n_k-1} \|P_{k,i} - P_{k,i-1}\|_2$$

$$\boxed{f_9 := \frac{1}{BB_{dia}} \cdot \sum_{k=0}^{F-1} len_k}$$ (full length over all traces)

$$\mu_{len} = \frac{1}{F} \cdot \sum_{k=0}^{F-1} len_k$$ (average length of each finger trace)

$$\boxed{f_{10} := \frac{1}{BB_{dia}} \cdot \mu_{len}}$$

$$\boxed{f_{11} := \frac{1}{BB_{dia}} \cdot \frac{1}{F} \cdot \sum_{k=0}^{F-1} \left(len_k - \mu_{len}\right)^2}$$ (standard deviation)

Implementation of the Pipeline of Gesture Recognition

Feature extraction *Standard deviation of distance to center*

Measurement of ”roundness” of a pattern

$$dist_{avg} = \frac{1}{F} \sum_{k=0}^{F-1} \frac{1}{n_k} \sum_{i=0}^{n_k-1} \|P_{k,i} - \mu\|_2$$

$$f_{13} := \frac{1}{BB_{dia}} \cdot \frac{1}{F} \sum_{k=0}^{F-1} \frac{1}{n_k} \sum_{i=0}^{n_k-1} \left(\|P_{k,i} - \mu\|_2 - dist_{avg} \right)^2$$

Implementation of the Pipeline of Gesture Recognition

Feature extraction *Standard deviation of distance to center*

Measurement of "roundness" of a pattern

$$dist_{avg} = \frac{1}{F}\sum_{k=0}^{F-1}\frac{1}{n_k}\sum_{i=0}^{n_k-1}\|P_{k,i}-\mu\|_2$$

$$f_{13} := \frac{1}{BB_{dia}}\cdot\frac{1}{F}\sum_{k=0}^{F-1}\frac{1}{n_k}\sum_{i=0}^{n_k-1}\left(\|P_{k,i}-\mu\|_2 - dist_{avg}\right)^2$$

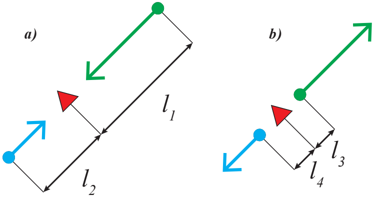
same computation per single finger trace with μ_k and $dist_{avg(k)}$

$$\mu_k = \frac{1}{n_k}\sum_{i=0}^{n_k-1}P_{k,i} \quad \text{and} \quad dist_{avg(k)} = \frac{1}{n_k}\sum_{i=0}^{n_k-1}\|P_{k,i}-\mu_k\|_2$$

$$f_{21} := \frac{1}{BB_{dia}}\cdot\frac{1}{F}\sum_{k=0}^{F-1}\frac{1}{n_k}\sum_{i=0}^{n_k-1}\left(\|P_{k,i}-\mu_k\|_2 - dist_{avg(k)}\right)^2$$

Implementation of the Pipeline of Gesture Recognition

Feature extraction *Average distance of start points and focus*



$$f_{14} := \frac{1}{BB_{dia}} \cdot \frac{1}{F} \sum_{k=0}^{F-1} \|P_{k,0} - \mu\|_2$$

Well suited for differentiation of *zoom in* and *zoom out* gestures

$$\frac{l_1+l_2}{2} \gg \frac{l_3+l_4}{2}$$

General: measure for expanding or reducing movement

Implementation of the Pipeline of Gesture Recognition

Feature extraction *Distance of synchronous occuring touch points*

(Use list of distances of synchronous occured points from recognition)

Normalized average tells how far multiple fingers are away from each other

$$f_{26} := \frac{1}{BB_{dia}} \cdot \frac{1}{G} \sum_{t=t_{start}}^{t_{end}} dist_t$$

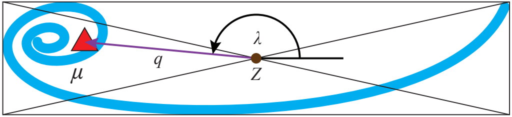
(G is the number of relevant measurements with more than 1 *finger)*

Standard deviation tells how ”parallel” fingers move

$$f_{27} := \frac{1}{BB_{dia}} \cdot \frac{1}{G} \sum_{t=t_{start}}^{t_{end}} \left(dist_t - f_{26}\right)^2$$

Implementation of the Pipeline of Gesture Recognition

Feature extraction *Eccentric main components*



Inspect vector q from center of bounding box Z to focus μ

Length of q as measure of eccentricity of the gesture

$$f_{24} := \frac{1}{BB_{dia}} \cdot \|\mathbf{q}\|_2, \text{ mit } \mathbf{q} = \mu - Z$$

Angle λ for direction of the main components

$$f_{25} := \lambda = \sphericalangle \mu Z (Z + \binom{1}{0})$$

Implementation of the Pipeline of Gesture Recognition

Feature extraction *Time & Speed*

Number of touch points of some trace $\sim$ execution time of this trace
(due to equidistant time sampling of the display)

$$f_{35} := \sum_{k=0}^{F-1} n_k$$

$\sim$ time to execute all touch traces sequently

Average speed as defined original $\frac{\delta distance}{\delta time}$

$$f_{43} := \sum_{k=0}^{F-1} \frac{\frac{1}{\rho_{Display}} \cdot len_k}{\tau_{k,n_k-1} - \tau_{k,0}} = \frac{1}{\rho_{Display}} \cdot \sum_{k=0}^{F-1} \frac{len_k}{\tau_{k,n_k-1} - \tau_{k,0}}$$

For independence of display resolution, normalize with $\rho_{Display}$
(Note: By normalizing with BB_{dia} the factor $\rho_{Display}$ would be cut)

Implementation of the Pipeline of Gesture Recognition

Feature extraction *Basic characteristics*

$$f_{12} := F$$ total number of used fingers for the gesture

$$f_{15} := \max_{t_{start} \leq t \leq t_{end}} \left(\sum_{k=0}^{F-1} \sum_{\substack{i=0 \\ \tau_{k,i}=t}}^{n_k-1} 1 \right)$$ maximum number of used fingers at the same time

$$f_{28} := \frac{1}{\rho_{Display}} \cdot BB_{dia}$$ differ same shape but different scaled gestures

$$f_{31} := \frac{BB_{Width}}{BB_{Height}}$$ propotions of gesture *(rather vertical or horizontal or compact)*

Implementation of the Pipeline of Gesture Recognition

Training

Progress so far:

1. User draws (multitouch) gesture, points are recorded
2. Lift last finger, compute all entries of feature vector

Now:

- Add class label to vector as ”last feature”
- Store all the taining data for building classifier later
- Build and train classifier

Implementation of the Pipeline of Gesture Recognition

Classification

Store build classifier object in file $\rightsquigarrow$ VNC App can use it

Nice option for testing classifier in *GesDe* App

Implementation of the Pipeline of Gesture Recognition

Classification

Store build classifier object in file $\rightsquigarrow$ VNC App can use it

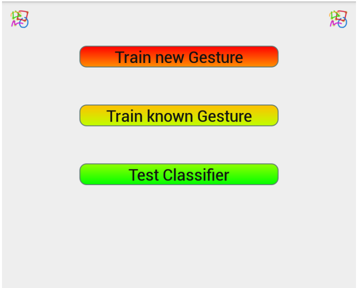
Nice option for testing classifier in *GesDe* App

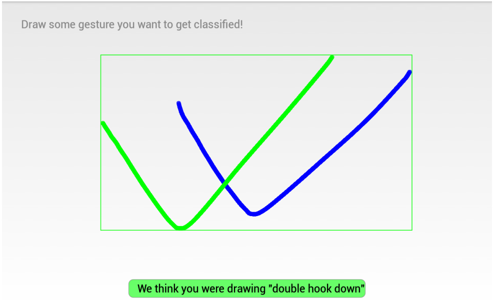
Implementation of the Pipeline of Gesture Recognition

Classification

Store build classifier object in file ⇝ VNC App can use it

Nice option for testing classifier in *GesDe* App





Survey

Main goals:

- $\Rightarrow$ **Find best classifier for VNC Application**
- $\Rightarrow$ **Rating of the quality of the features**

Secundary goals:

- $\rightarrow$ Analyse, whether the gesture recognition rate depends on the experience with touch devices
- $\rightarrow$ Analyse, whether gesture recognition depends on the age

Survey

Main goals:

- ⇒ **Find best classifier for VNC Application**
- ⇒ **Rating of the quality of the features**

Secundary goals:

- → Analyse, whether the gesture recognition rate depends on the experience with touch devices
- → Analyse, whether gesture recognition depends on the age

Need for representative participant distribution

age [years]	number of participants	fraction [%]
5-17	4	16
18-39	10	40
39-60	7	28
61+	4	16
Σ	25	100

Survey

Environment setup

Studie application for survey as modified brach of *GesDe* app

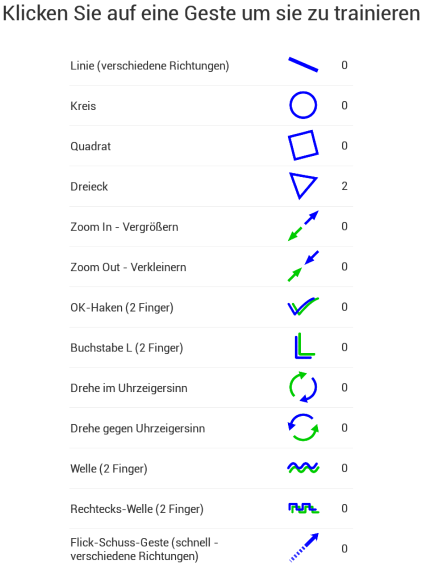
Advantages:

- automatic supervision of the number of samples per gesture
- automatic generation of new dataset for each participant
- request for age and experience per GUI and link information with dataset
- suited for unsupervised participation due to datailed instruction
- suited for digital spreading due to built-in return of dataset per mail

Survey

Content setup

- force 20 repetitions
  of each single gesture
- animate participant
  to vary size and accuracy
- idealised previews
- German language
- choice of gestures:
  - some standard gestures
  - always bundle of similar pattern



Results of the survey and conclusion

Results of the survey and conclusion

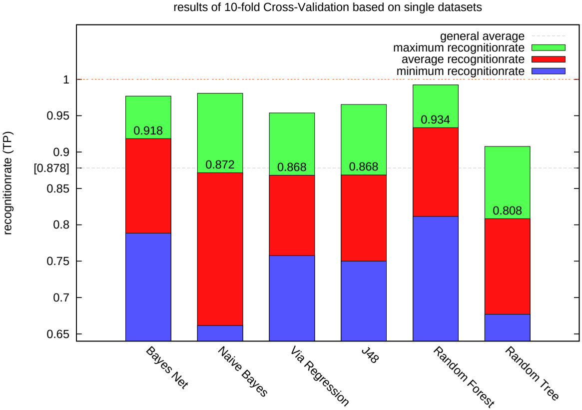
Choise of classifiers for testing

1. Bayes Net
2. Naive Bayes
3. Classification via regression
4. J48
5. Random Forest
6. Random Tree

All parameters of adjustable classifiers are default values

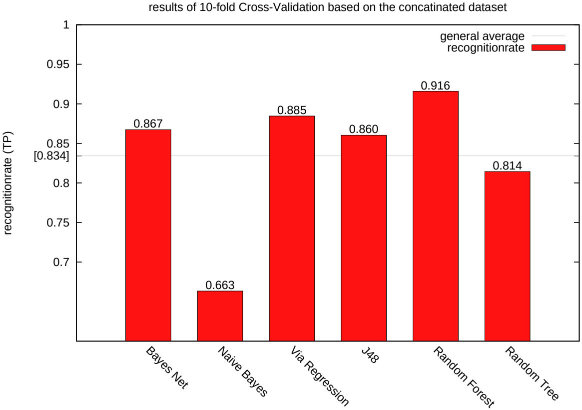
Results of the survey and conclusion

10-fold Cross-Validation based on 28 single datasets



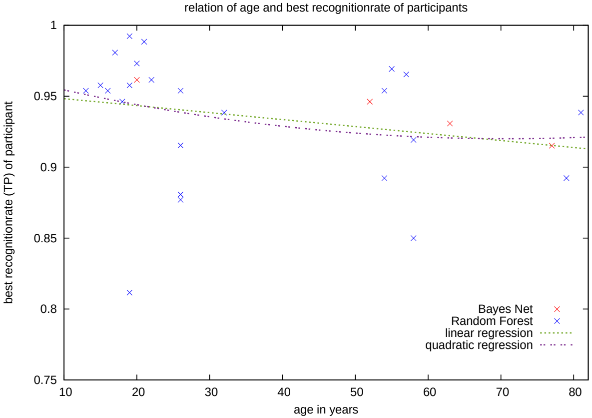
Results of the survey and conclusion

10-fold Cross-Validation based on one merged dataset



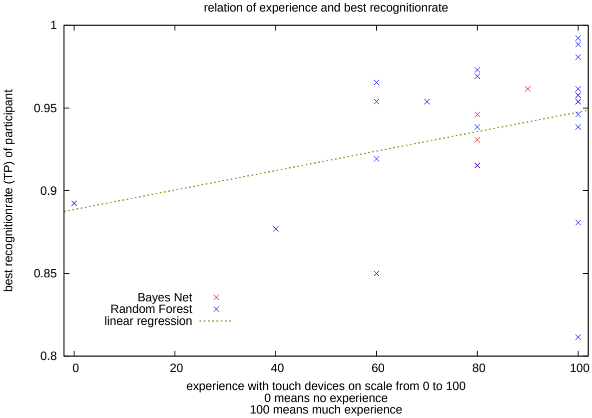
Results of the survey and conclusion

Relation of age of participants to their best recogintionrate



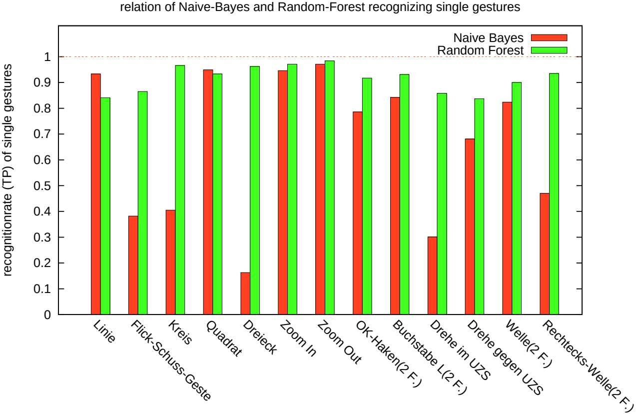
Results of the survey and conclusion

Relation of experience of participants to their best recognition rate



Results of the survey and conclusion

Comparison of the recognitionrates of different gestures



Summary & Outlook

Summary

- Multitouch transfer per extended VNC
- Issue of belonging solved by flagfields
- Embedded gesture recognition for smart control
- Separate application for training the classifier and introduce gestures
- Survey for comparing different classifiers
- Random Forest algorithm reached best result
- In general recognitionrate is independent of age
- Experience with touch devices influences the recognition
- Some differences on recognitionrate of different gestures

Outlook

Multitouch VNC

- Dynamic mapping from classlabel to reaction
- Build overview of existing gesture-to-reaction-mapping
- VNC in general:
  introduce better compressing algorithms for higher framerate

Gesture recognition

- More or better features to differentiate the worse differentiated gestures
- Feature selection on existing features
- Consider other gesture schemas e.g.numbers or letters
- Realtime recognition
- Detect non connected gestures

Discussion

Discussion

Thanks for your attention