Learning Multi-Touch Interfaces: Gesture Recognition and Automated Interaction Design

Q. Debard • C. Wolf • S. Canu • J. Arné • J. S. Dibangoye

LIRIS – Itekube – LITIS – CITI-Lab

14th March, 2019

Overview

- **Background and motivations**
  - Touch interfaces
  - Interaction protocols
- **Gesture recognition**
  - Data understanding
  - Models and results
- **Automated interaction design**
  - Formalization and early leads
- **What's next ?**

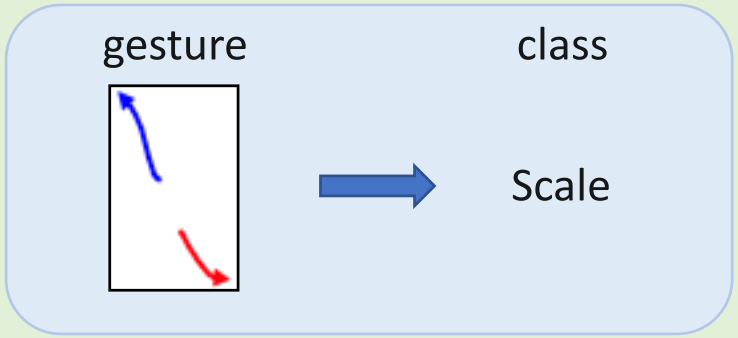
Pros:

- More natural interactions than usual mouse and keyboard
- Simple interactions well designed: pinch, rotate, translate, tap
- Can easily be multi-user

Cons:

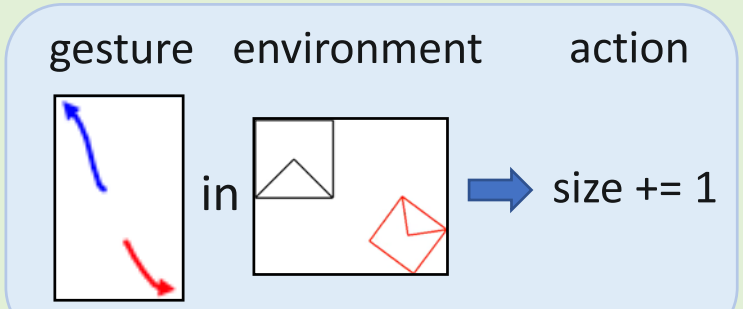
- Generally less precise than mouse and keyboard
- Hard to define interaction protocols for complex uses

Two complementary problems



Recognition of discrete gestures

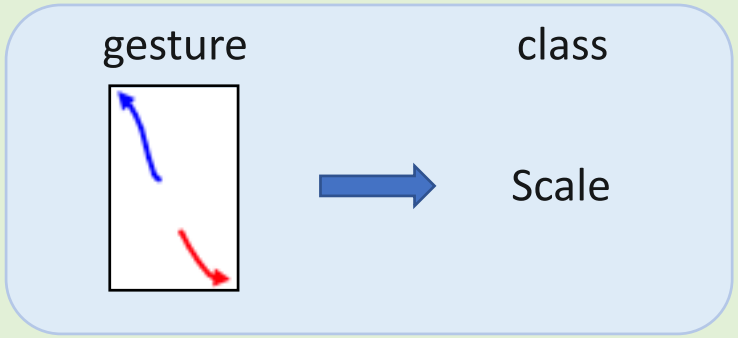
→ Classification, supervised learning



Continuous coupling of gestures and actions

→ Reinforcement Learning

Gesture recognition



Recognition of discrete gestures

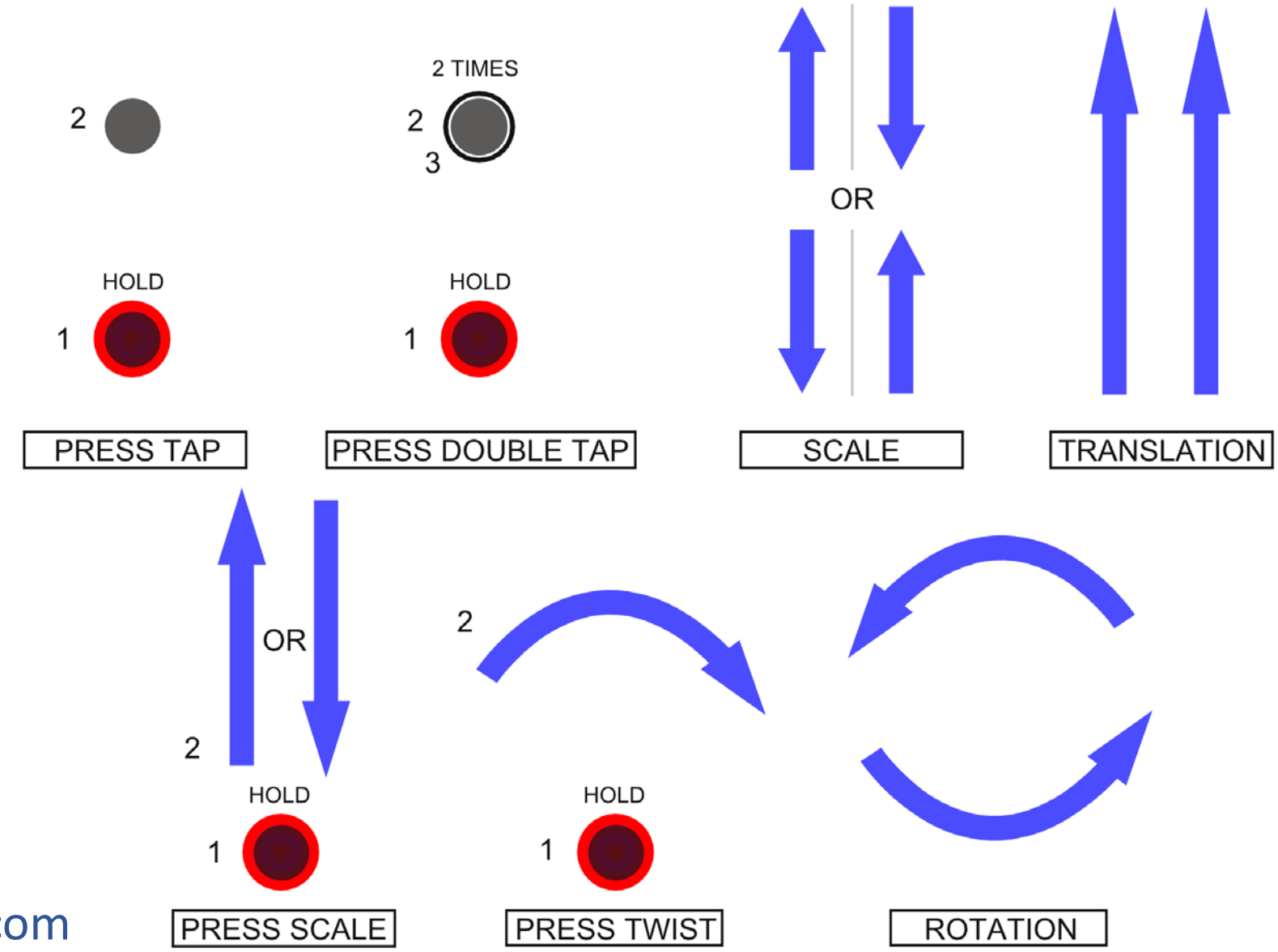
→ Classification, supervised learning

Continuous coupling of gestures and actions

→ Reinforcement Learning

Gesture recognition – Data understanding

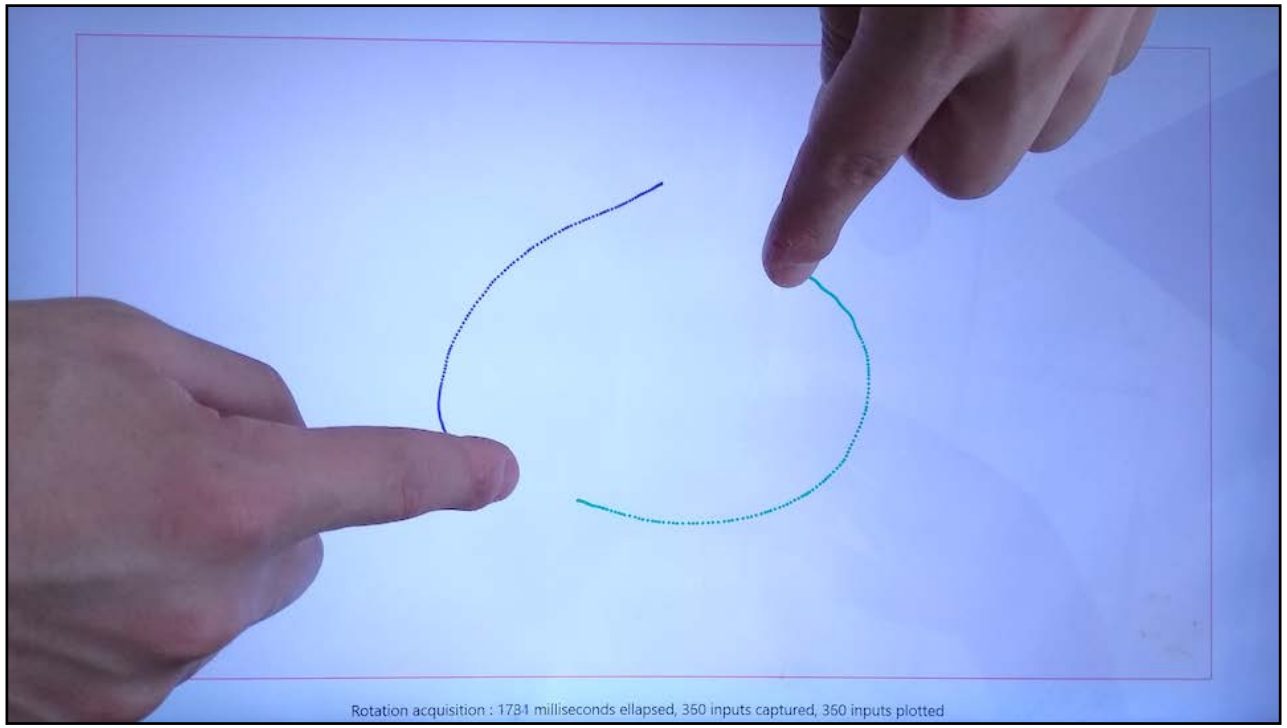
- 7 Classes
- 27 Users
- 6591 Gestures
- High variability



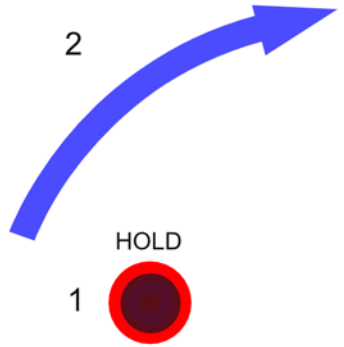
- Publicly available at:

  http://itekube7.itekube.com

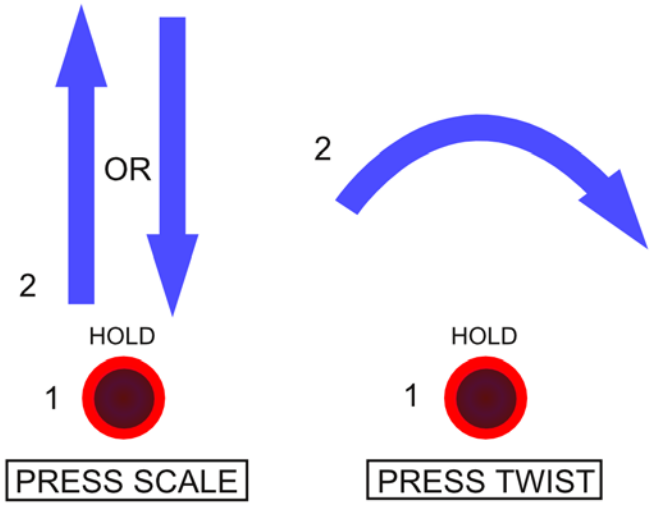
Gesture recognition – Data understanding



In reality:

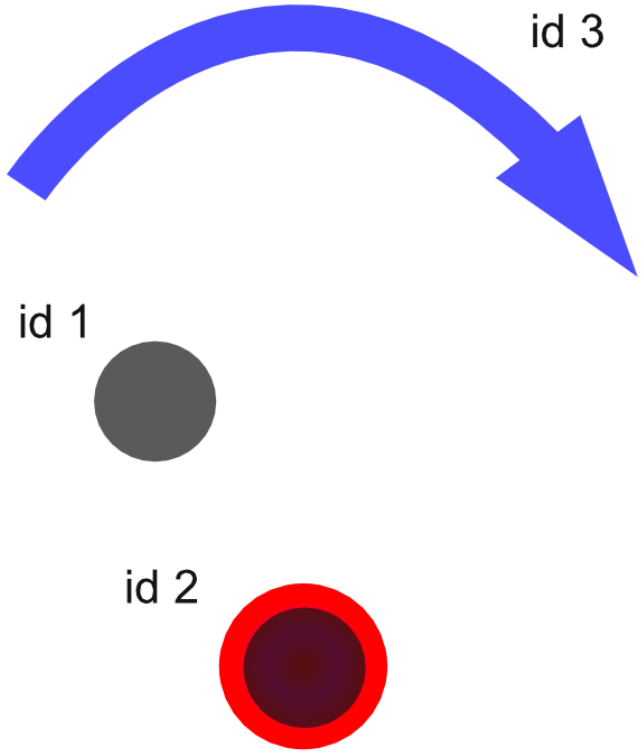


Press Scale or Press Twist?



Trajectories as sequences of (x,y) couples

GESTURE



CONTACT MATRIX

	t_1	t_2	t_3	t_4	t_5	t_6	…
id 3	(0,0)	(0,0)	$(x,y)_{3,3}$	$(x,y)_{3,4}$	$(x,y)_{3,5}$	$(x,y)_{3,6}$	…
id 2	(0,0)	$(x,y)_{2,2}$	$(x,y)_{2,3}$	$(x,y)_{2,4}$	$(x,y)_{2,5}$	$(x,y)_{2,6}$	…
id 1	$(x,y)_{1,1}$	$(x,y)_{1,2}$	$(x,y)_{1,3}$	$(x,y)_{1,4}$	(0,0)	(0,0)	…

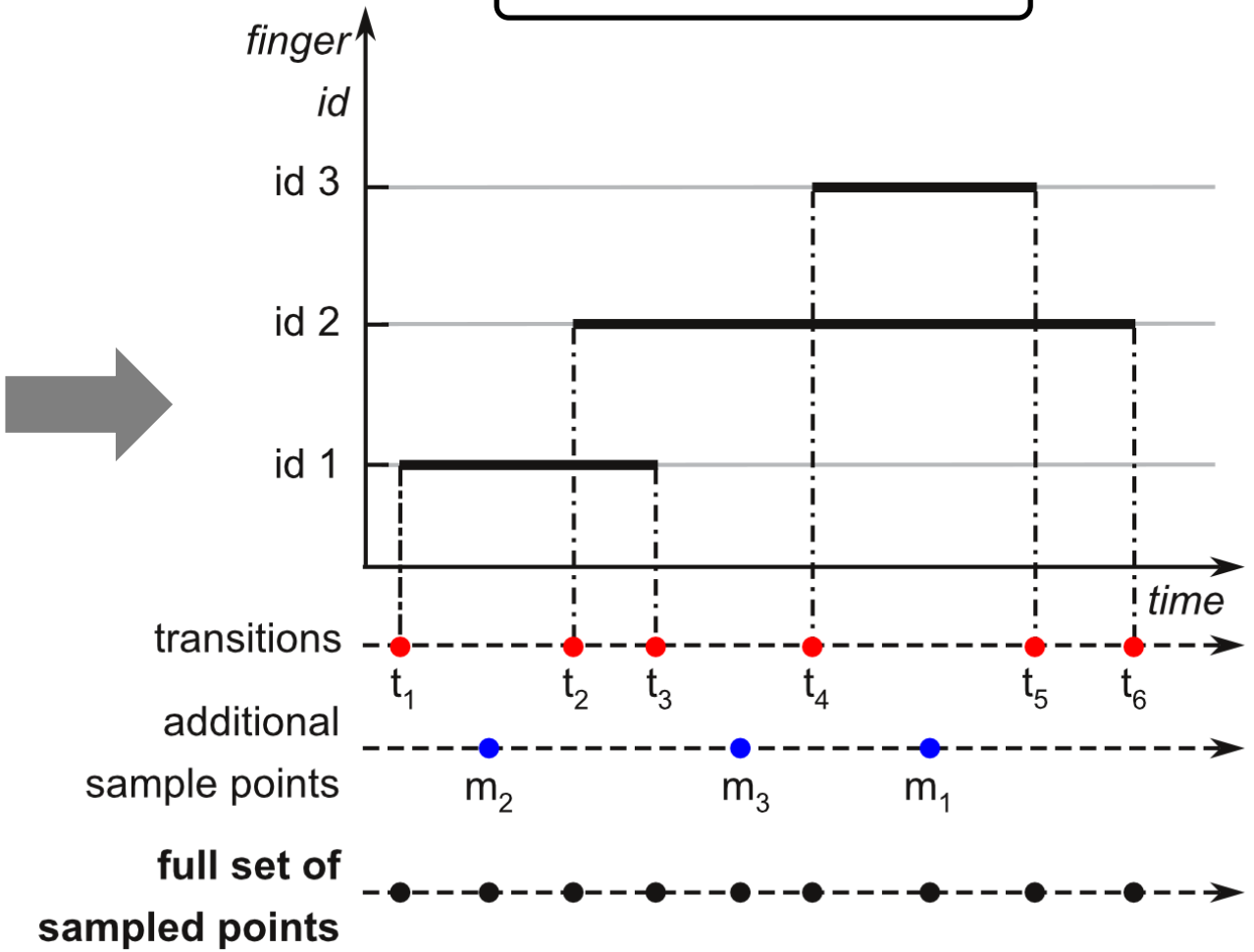
Raw Input → Data Processing → Classifier → Gesture

Gesture recognition – Data understanding

CONTACT MATRIX

	t_1	t_2	t_3	t_4	t_5	t_6	
id 3	(0,0)	(0,0)	$(x,y)_{3,3}$	$(x,y)_{3,4}$	$(x,y)_{3,5}$	$(x,y)_{3,6}$	...
id 2	(0,0)	$(x,y)_{2,2}$	$(x,y)_{2,3}$	$(x,y)_{2,4}$	$(x,y)_{2,5}$	$(x,y)_{2,6}$	...
id 1	$(x,y)_{1,1}$	$(x,y)_{1,2}$	$(x,y)_{1,3}$	$(x,y)_{1,4}$	(0,0)	(0,0)	...

DYNAMIC SAMPLING



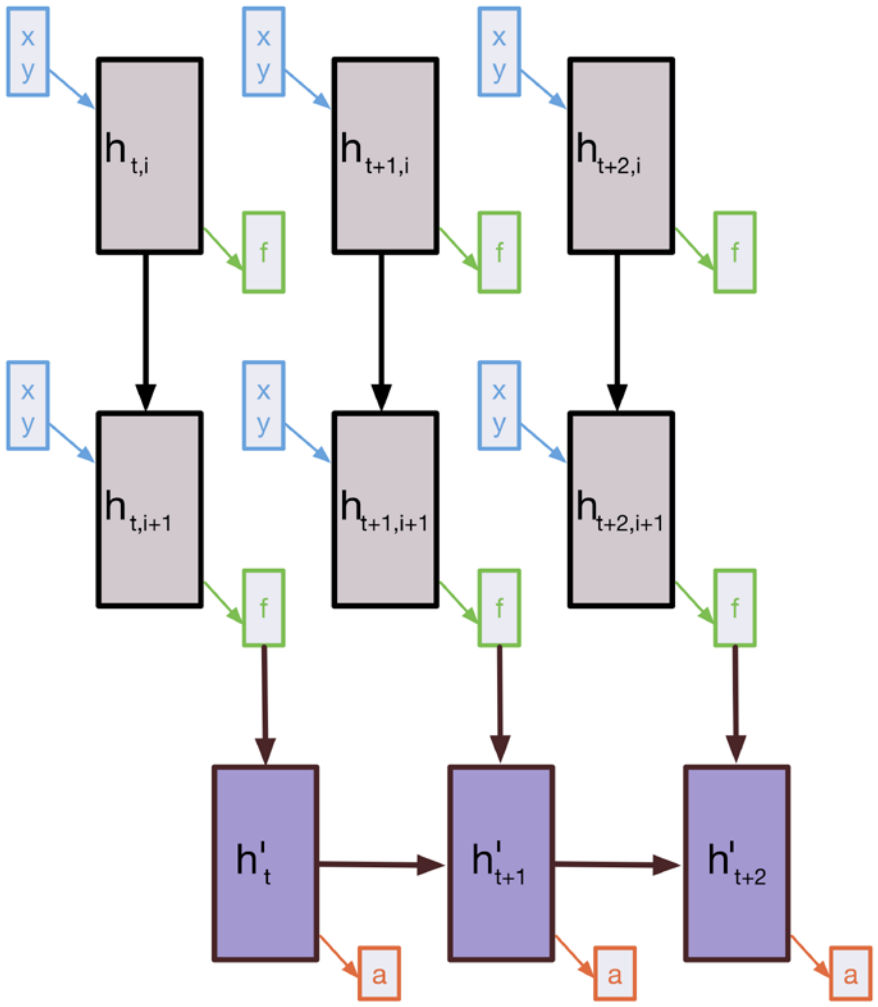
Raw Input → **Data Processing** → Classifier → Gesture

2 classical architectures:

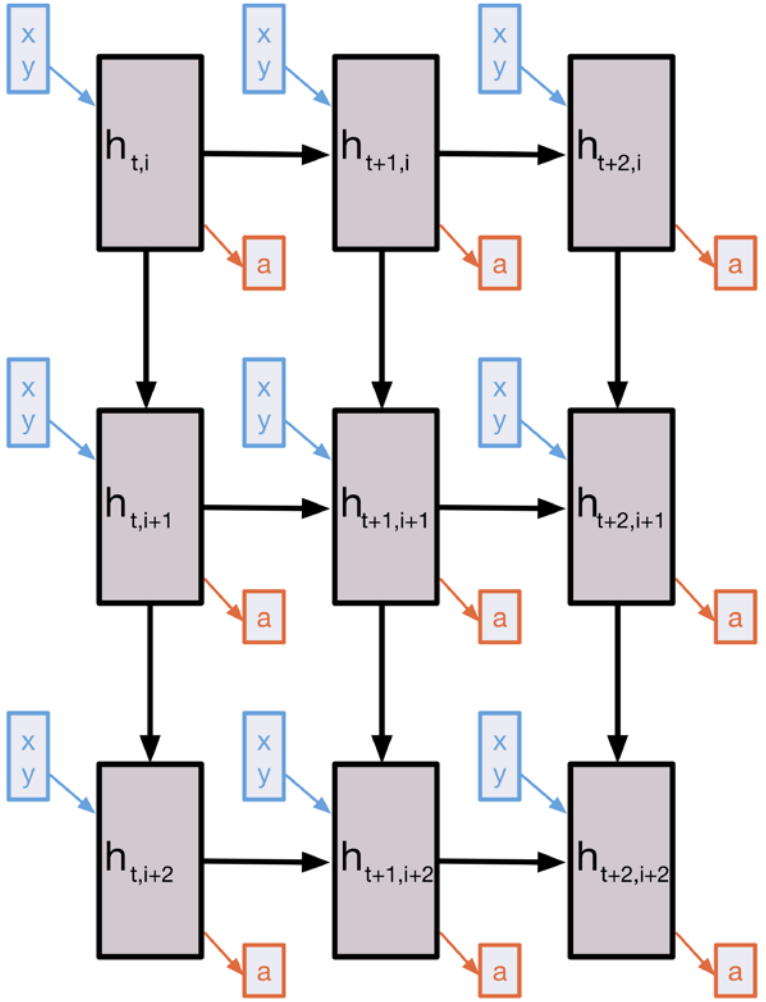
- Recurrent Neural Networks (RNNs)
- Convolutional Neural Networks (CNNs)

For sequential problems, RNNs are traditionally used

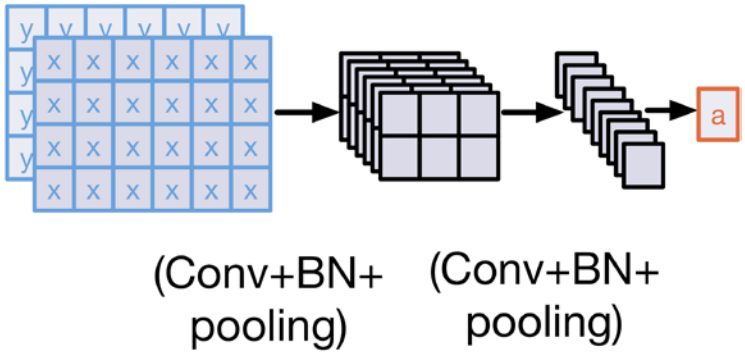
Gesture recognition – Models and results



LSTM



2D-LSTM



CNN

Raw Input → **Data Processing** → **Classifier** → Gesture

Leave-One-Subject-Out Cross Validation protocol

Methods	Sampling	Data augmentation	Accuracy
LSTM	-	X	58.71
LSTM	Dynamic	X	73.10
2D-LSTM	-	X	60.01
2D-LSTM	Dynamic	X	87.72
Convolutional model	-	-	65.96
Convolutional model	Dynamic	X	**89.96**

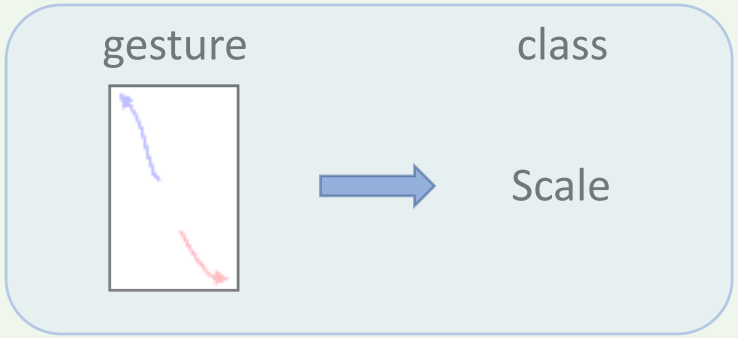
CNNs do perform better for our task

Learning to recognize touch gestures: recurrent vs. convolutional features and dynamic sampling

13th IEEE Conference on Automatic Face and Gesture Recognition (FG 18)

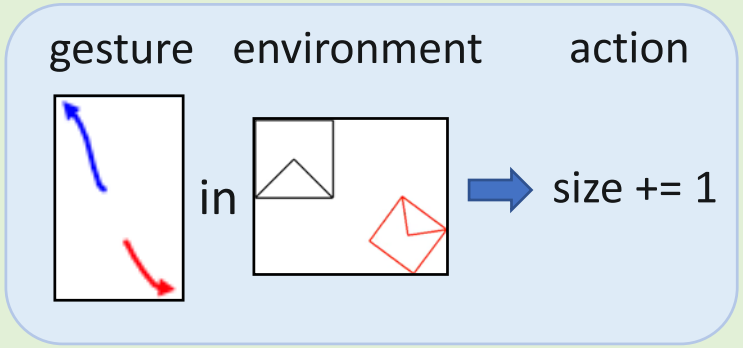
Q. Debard, C. Wolf, S. Canu and J. Arné

Automated interaction design



Recognition of discrete gestures

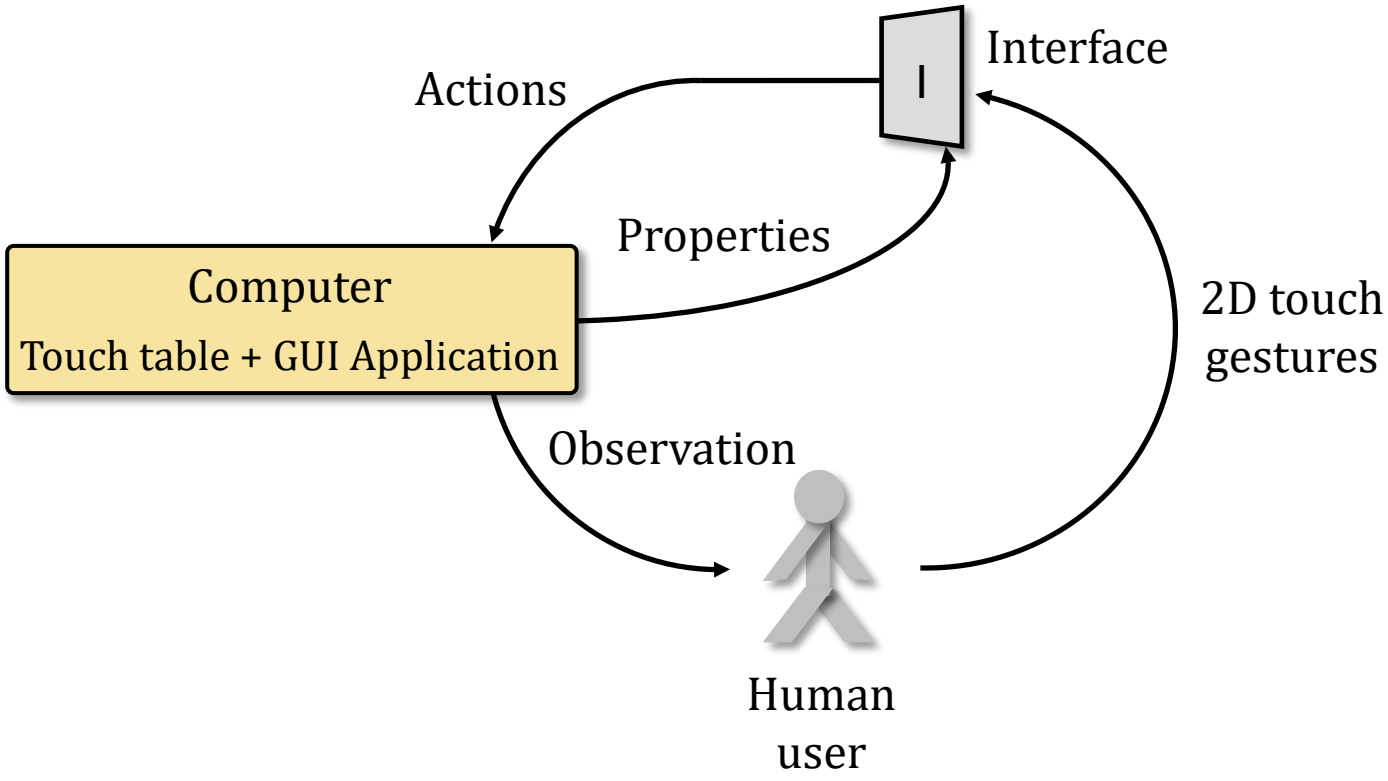
→ Classification, supervised learning



Continuous coupling of gestures and actions

→ Reinforcement Learning

Automated interaction design – Formalization and early leads



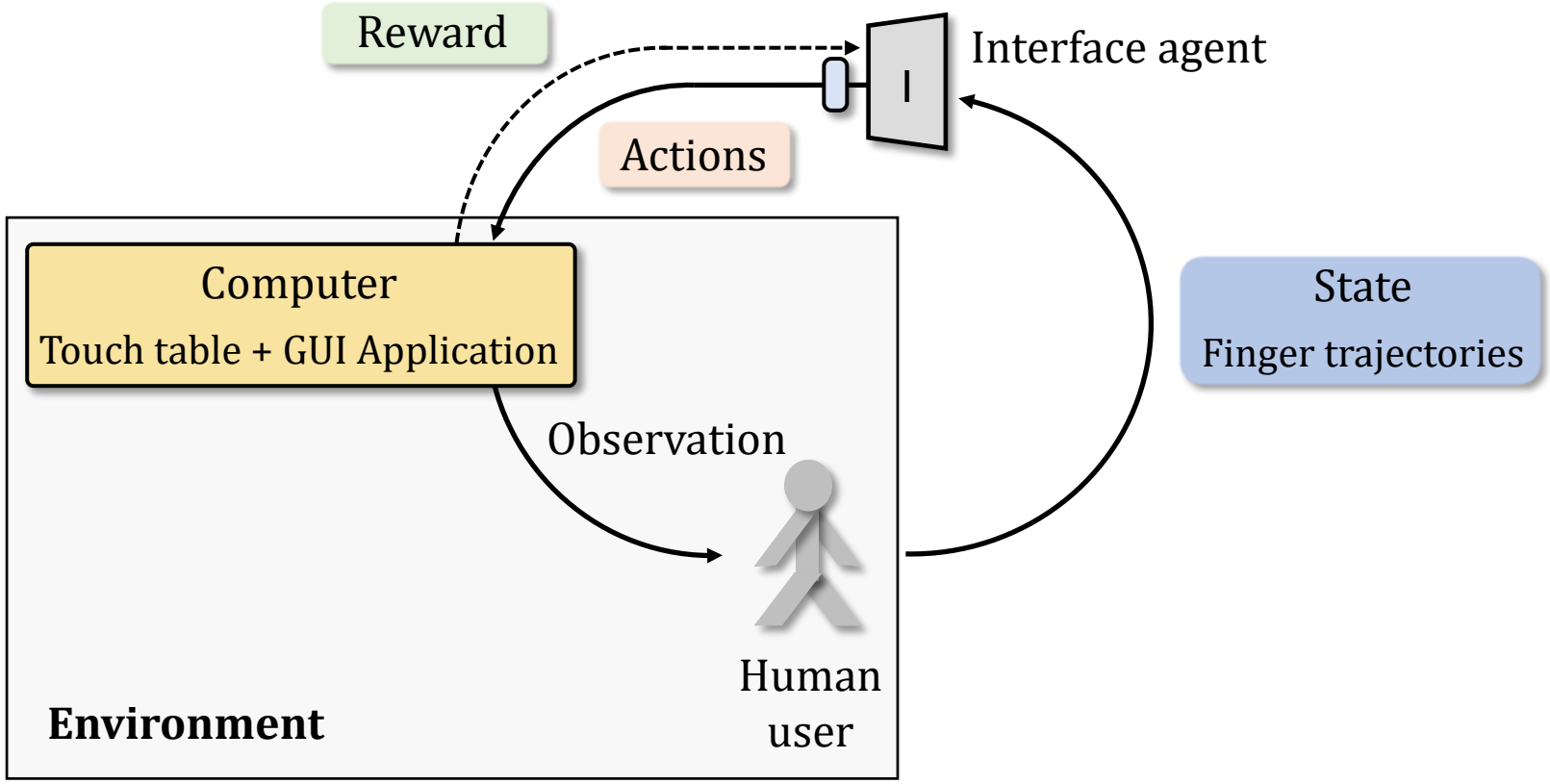
An interaction protocol can be seen as a function with:

Two inputs
- User's finger trajectories
- An observation of the application

One output
- Actions to be performed in the application

This function must optimize
- Ease of use
- Intuitive handling
- Precision
- Expressivity

This can be formulated as a Reinforcement Learning (RL) problem:



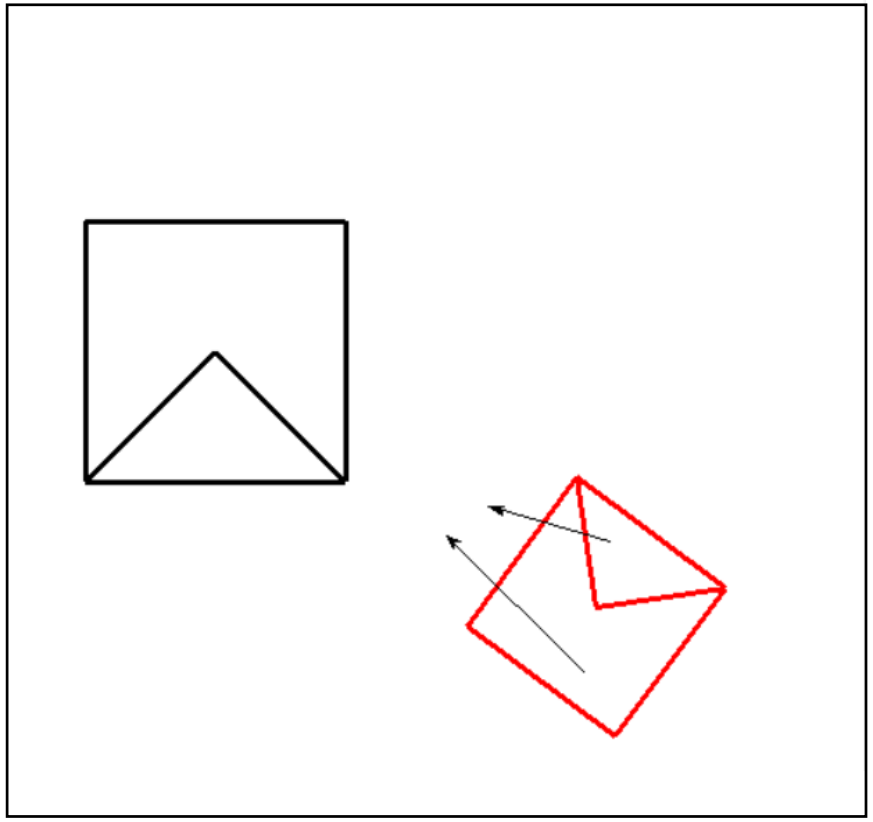
All this is just pre-training!

Once pre-trained, the interface agent will still be learning… From human users.

Toy environment: 2D manipulation

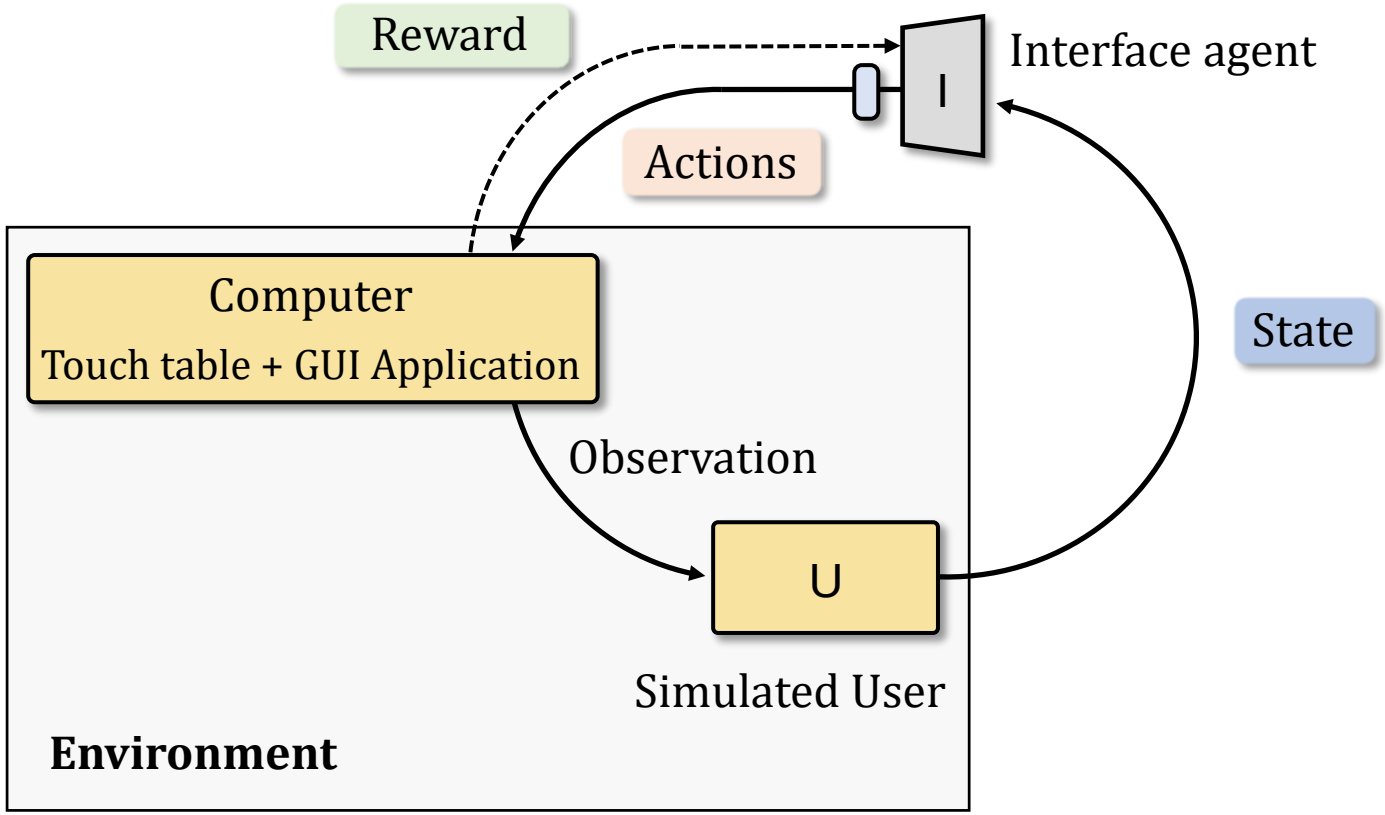
Objective: relearn the two-finger Pinch/Rotate/Translate interface

- Red shape: object to be manipulated
- Black shape: target position
- Arrows : user trajectories

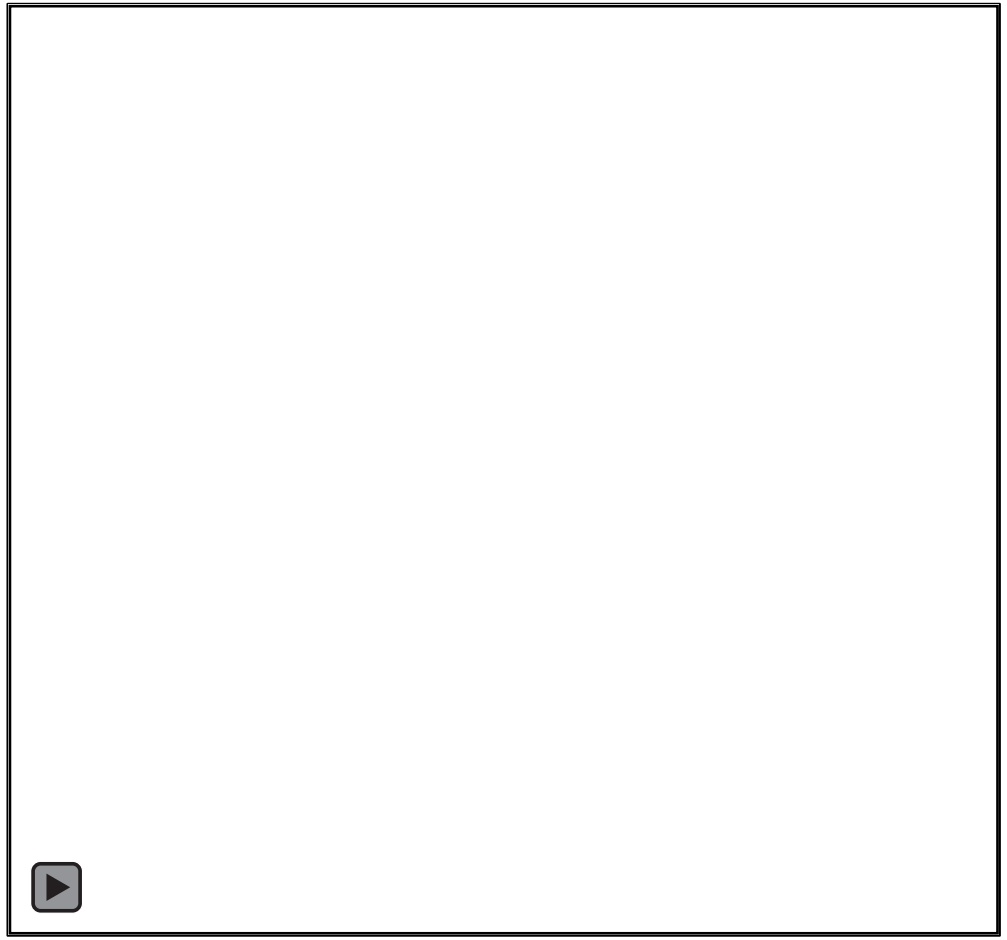


Problems:

- What we want to optimize on is not measurable
- We can not have real data for each state of the environment



After being trained for about 500 episodes:



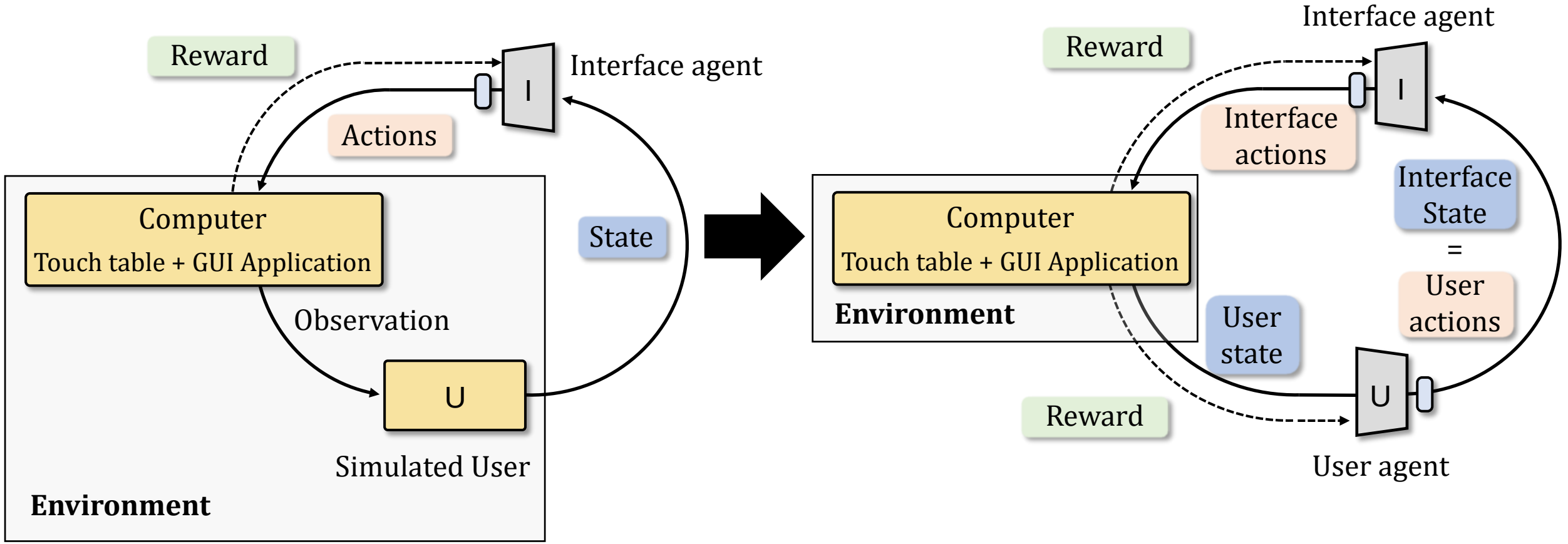
What's next?

As long as we know the optimal interaction protocol
=> We can analytically simulate an optimal user

What about 3D environments?

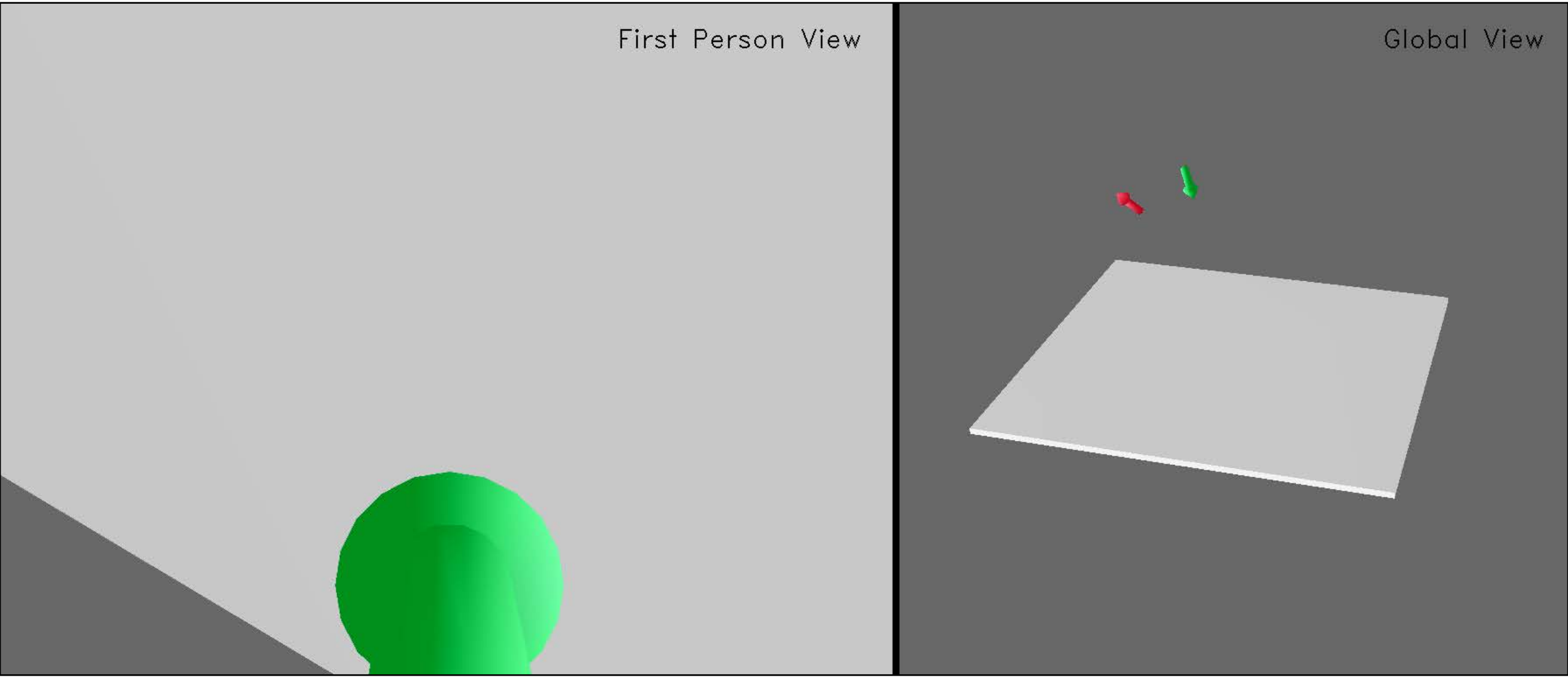
Ill-posed problem, trying to find a mapping from 2D to 3D space
=> No analytic solution to the user!

So… What can we do when we cannot analytically simulate the user ?



Also learn the user!

Our current 3D navigation environment:



Thank you for your attention

Gesture recognition article:

Learning to recognize touch gestures: recurrent vs. convolutional features and dynamic sampling,

13th IEEE Conference on Automatic Face and Gesture Recognition,

Q. Debard, C. Wolf, S. Canu and J. Arné

Itekube-7: http://itekube7.itekube.com

quentin.debard@insa-lyon.fr

https://perso.liris.cnrs.fr/quentin.debard/