Reconnaître un pattern de points de contact sur une surface tactile

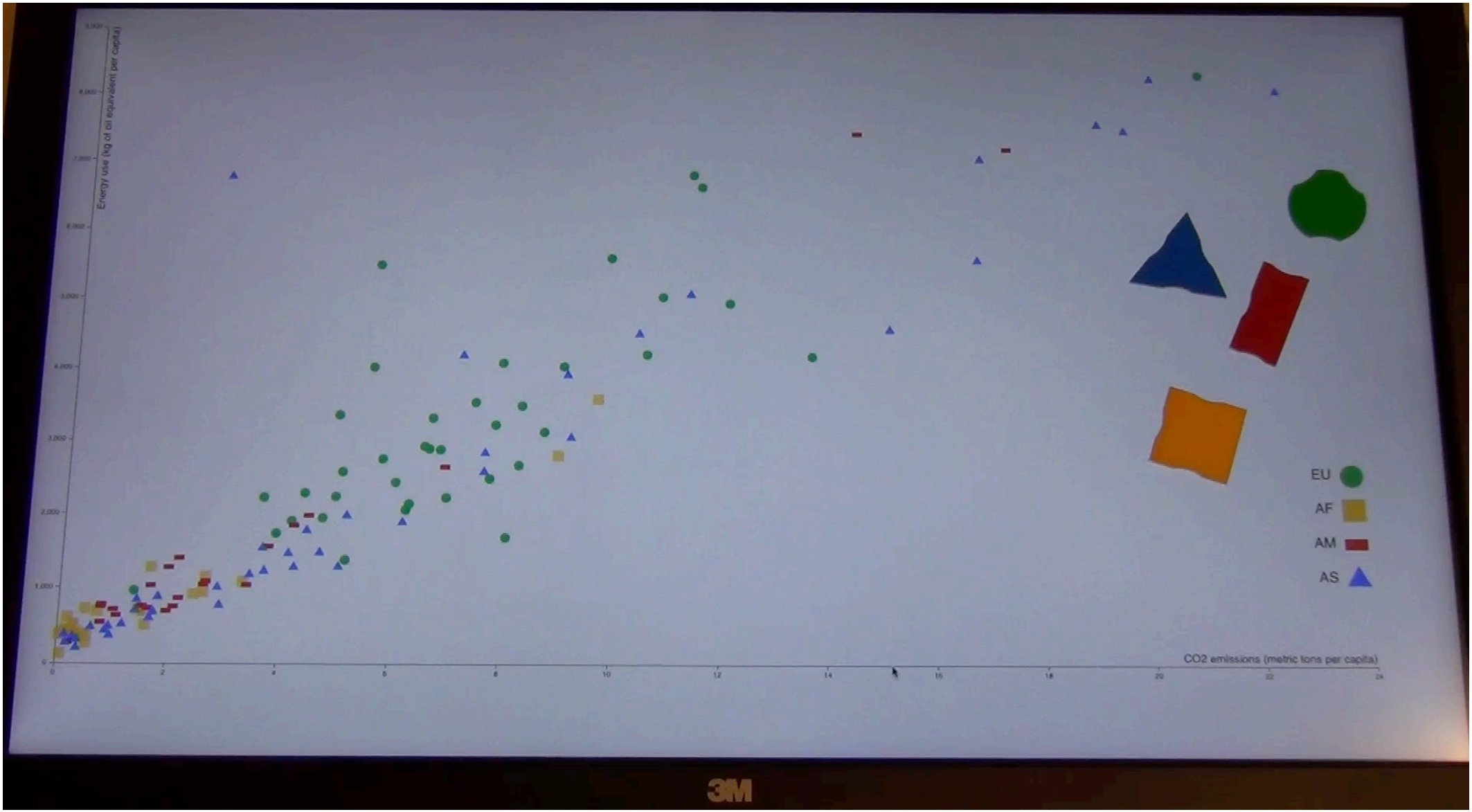
Multi-Touch et Machine Learning

Journée IHM-IA '17
Caroline Appert

TouchTokens

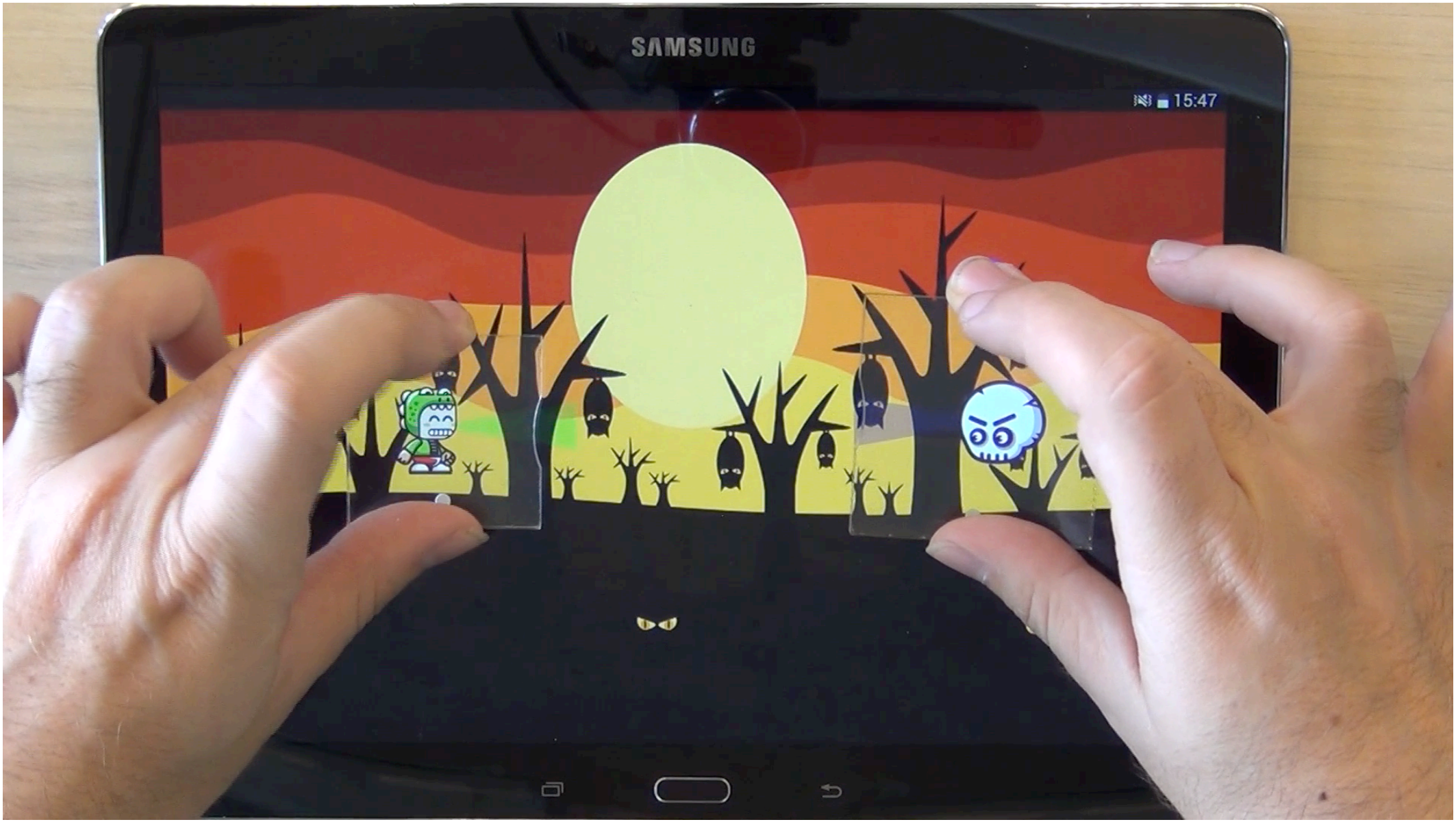
R. Morales Gonzalez, C. Appert, G. Bailly and E. Pietriga. CHI '16.
TouchTokens: Guiding Multi-Touch Patterns with Passive Tokens.

TouchTokens make it possible
to easily build interfaces that
combine gestures and tangibles



Data visualization

Tangible magic lens

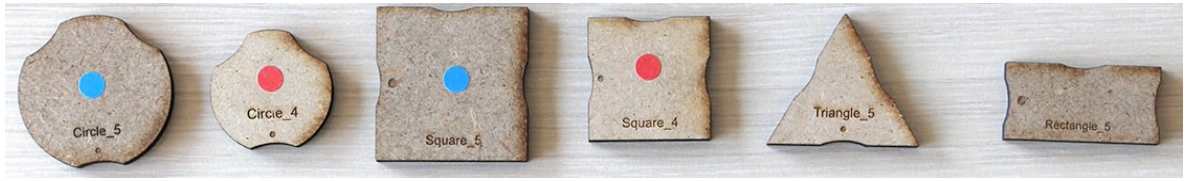


Character controllers in a game

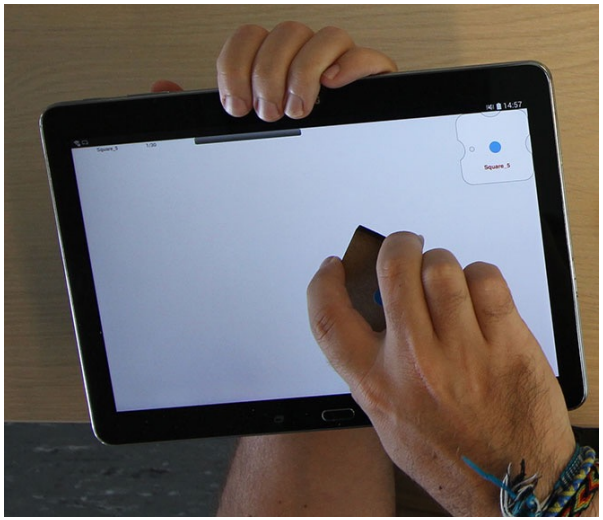
Access control

TouchTokens

Passive tokens indicating how to grasp them

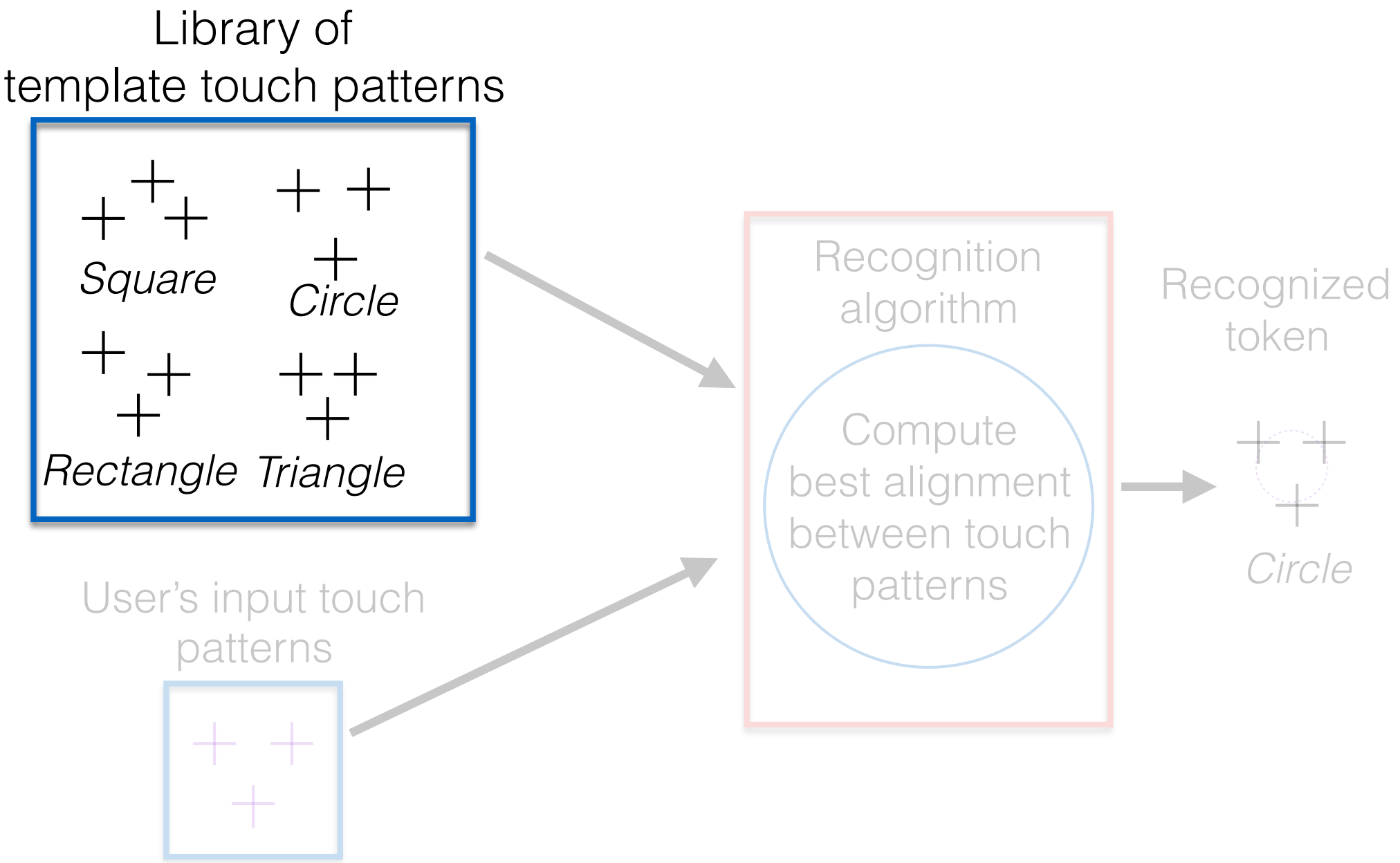


Tactile surface

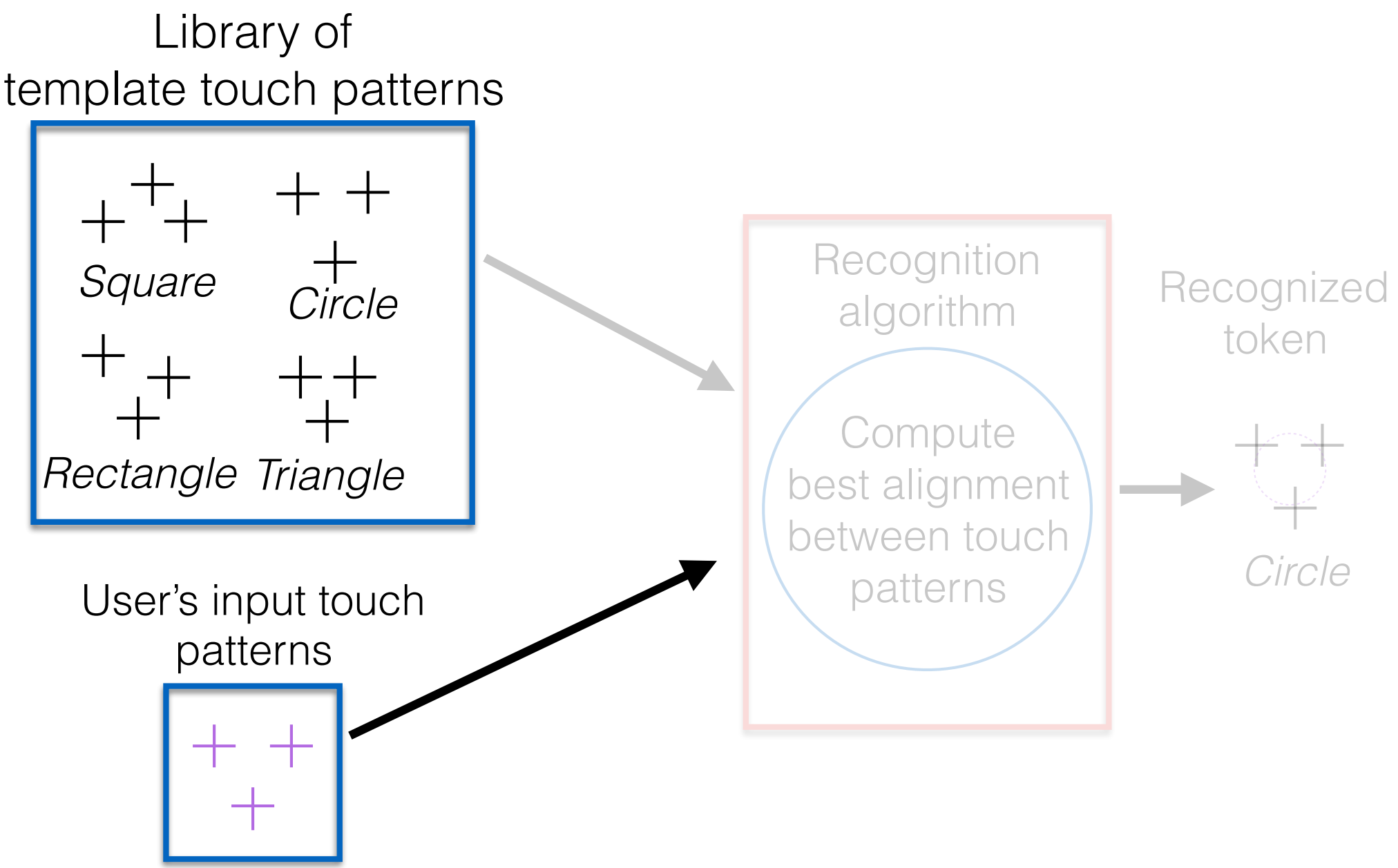


Touch pattern recognizer

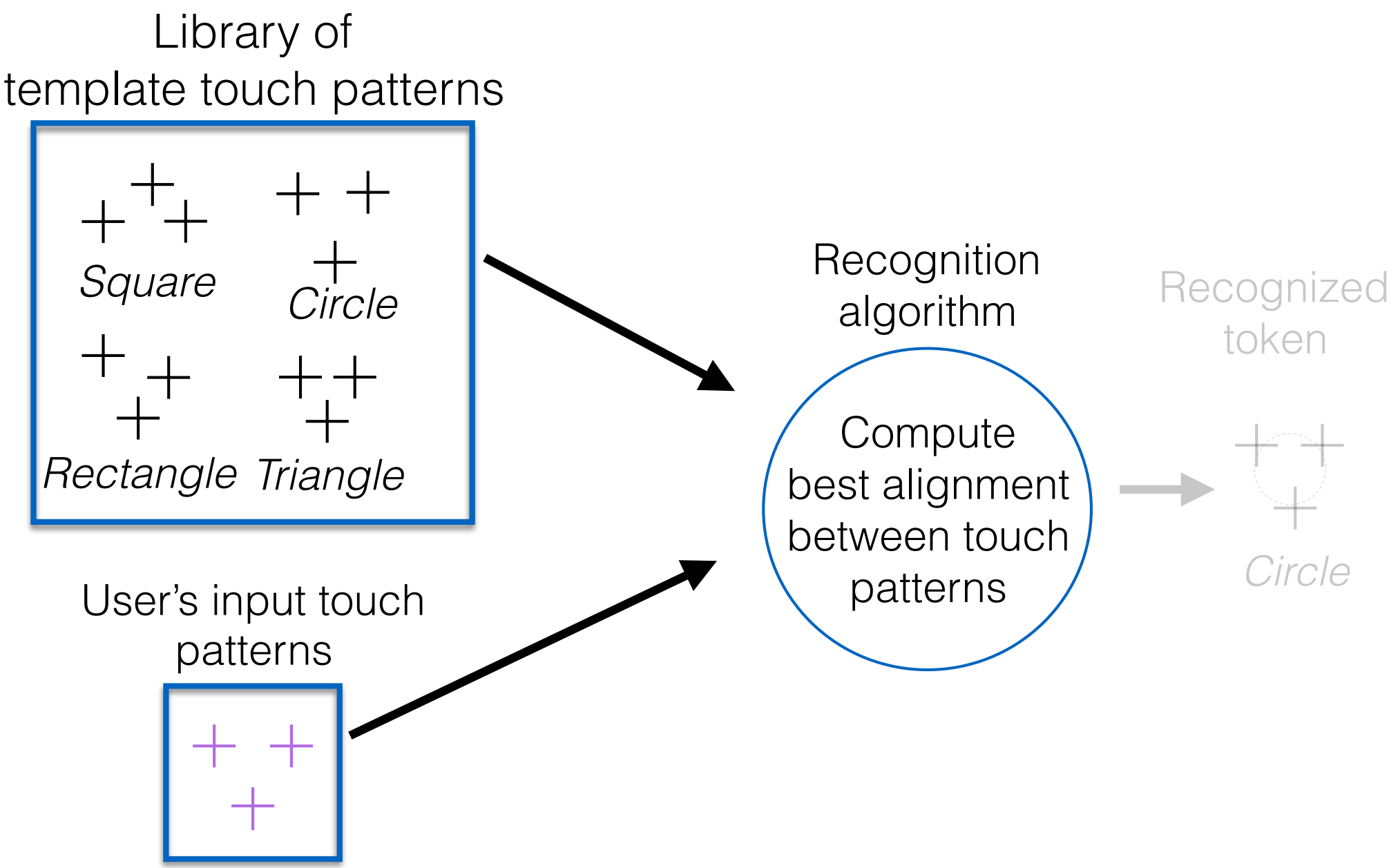
Touch pattern recognizer



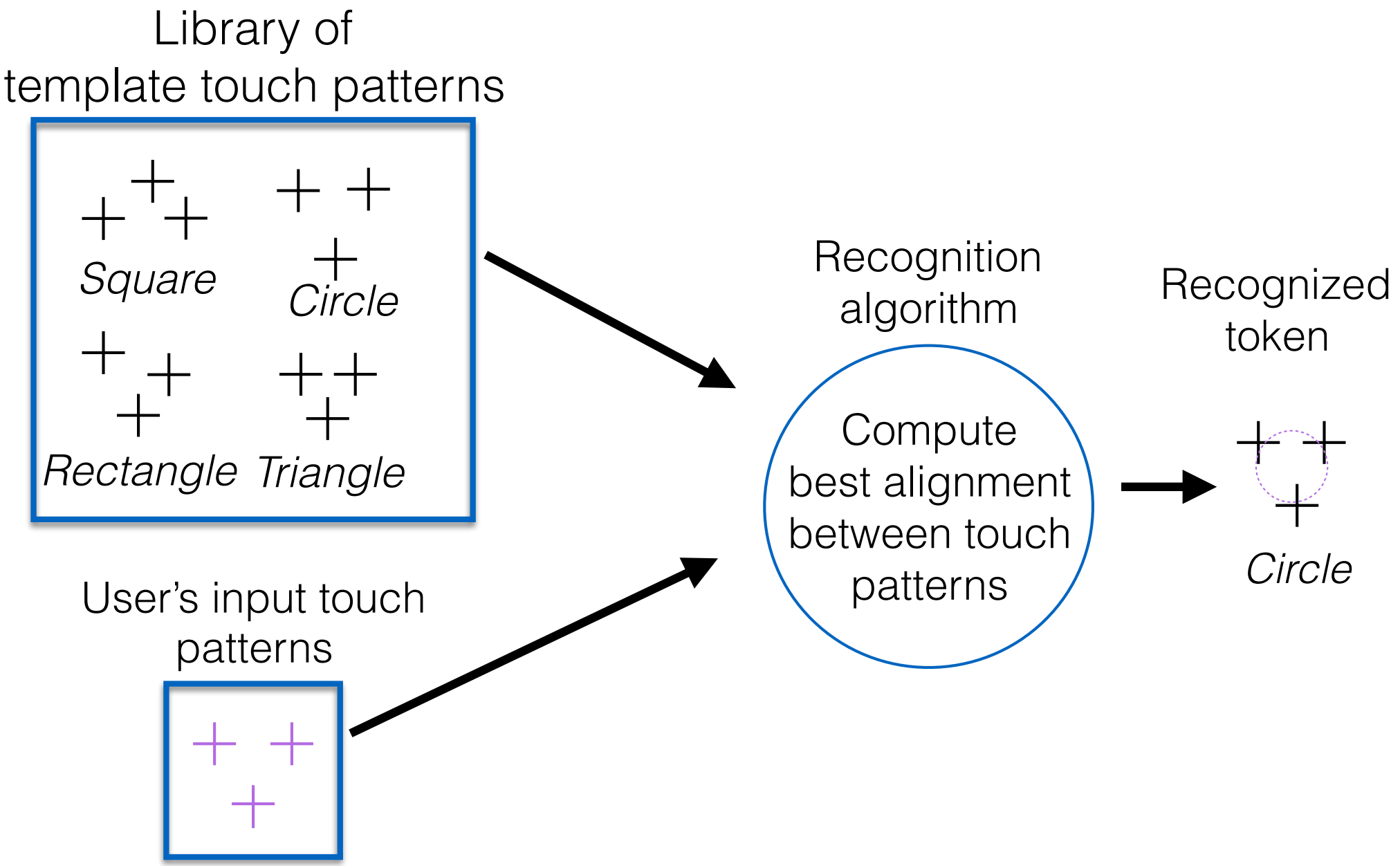
Touch pattern recognizer



Touch pattern recognizer

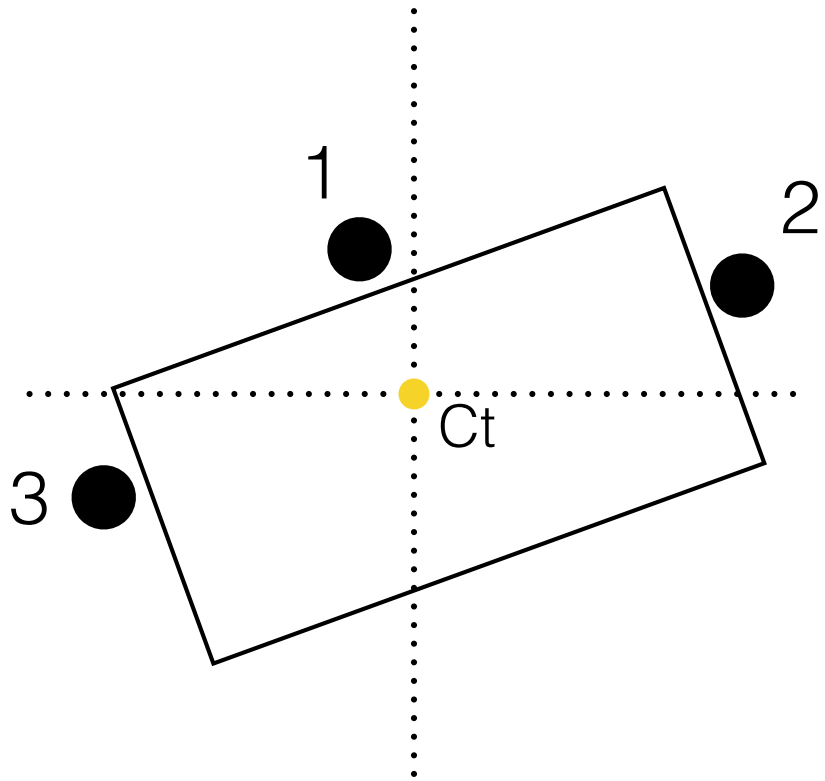


Touch pattern recognizer



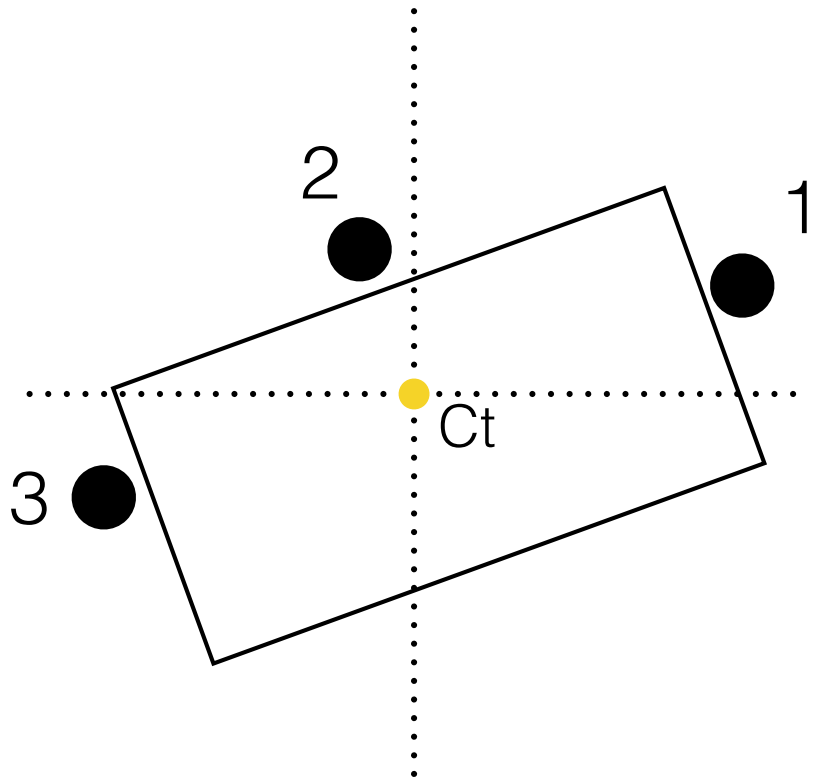
Best alignment algorithm

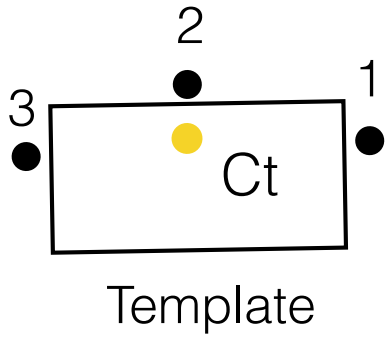
Template



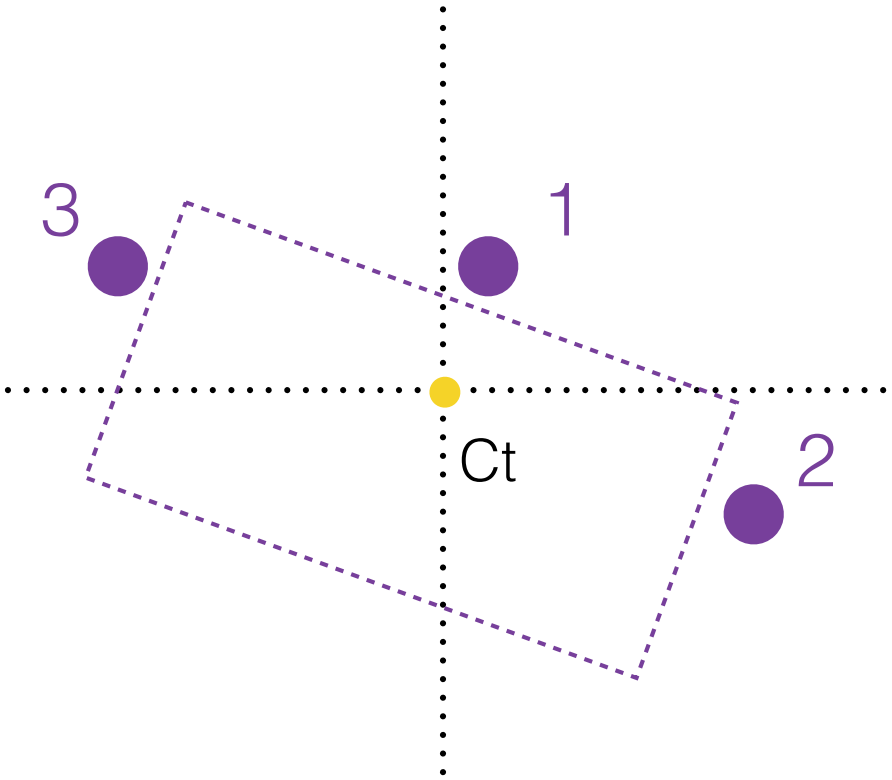
Best alignment algorithm

Template

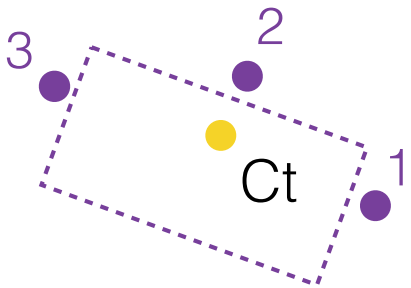




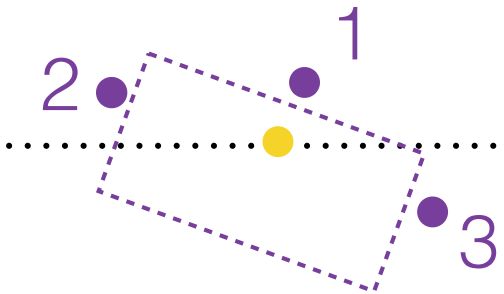
Input

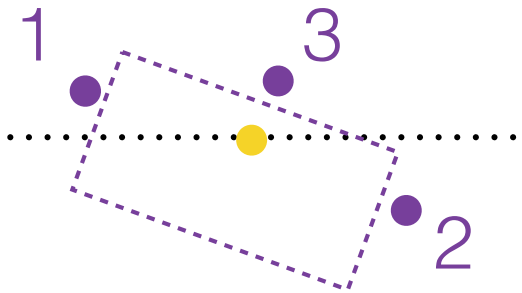


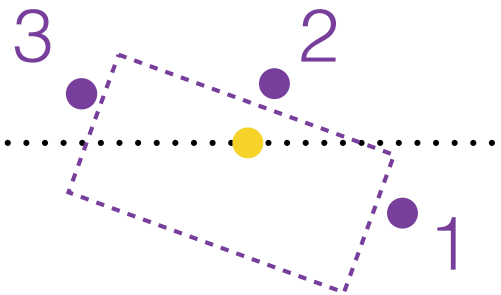
permutations

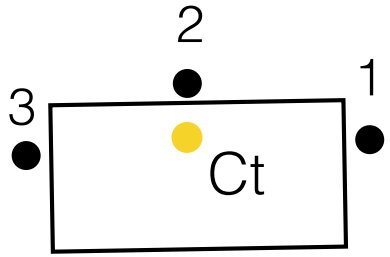


input



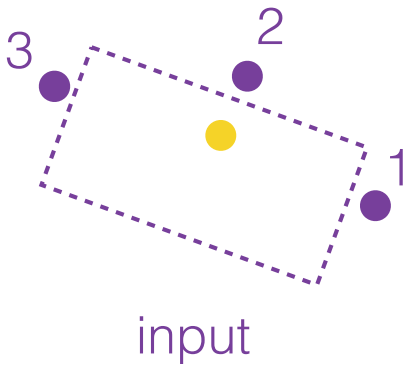




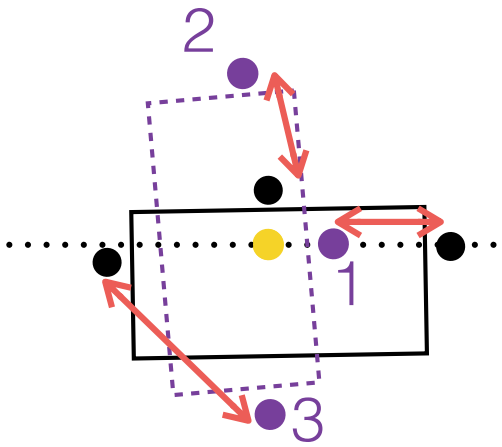


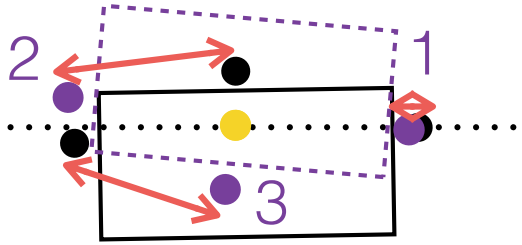
Template

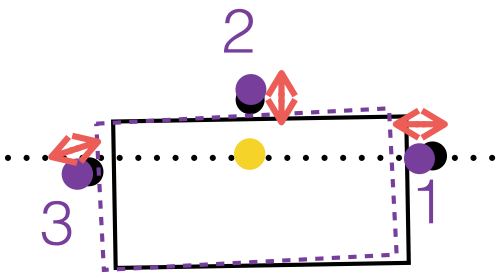
permutations

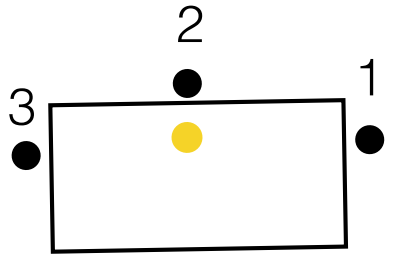


input



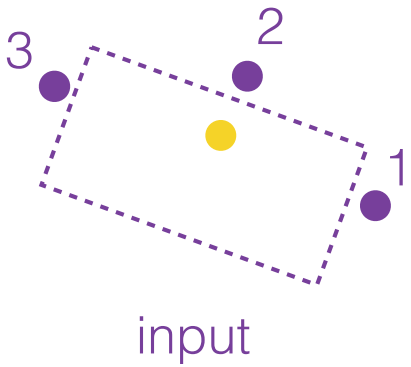


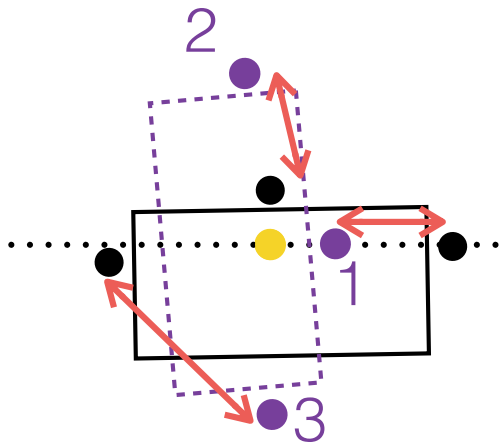


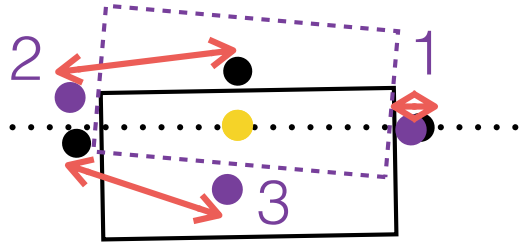


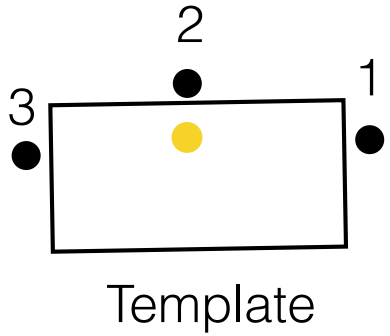
Template

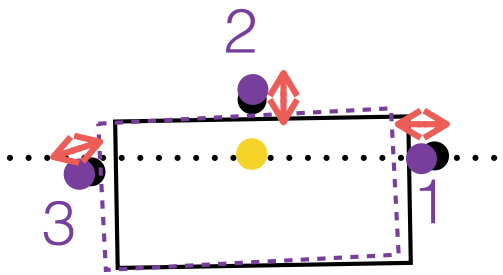
permutations







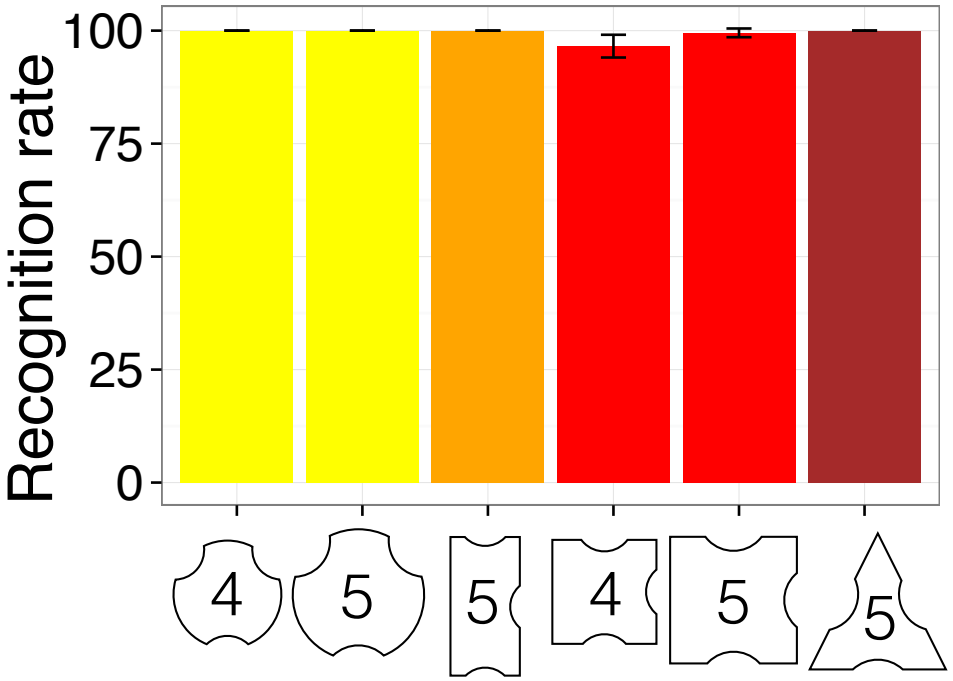




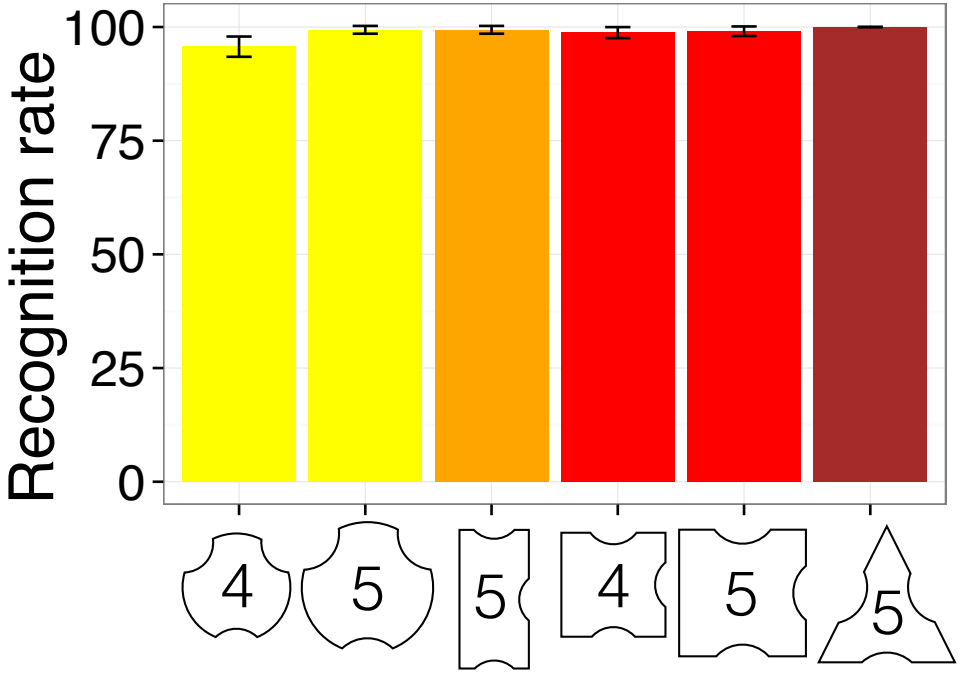
Best Alignment

Results

Tablet: 99.3%

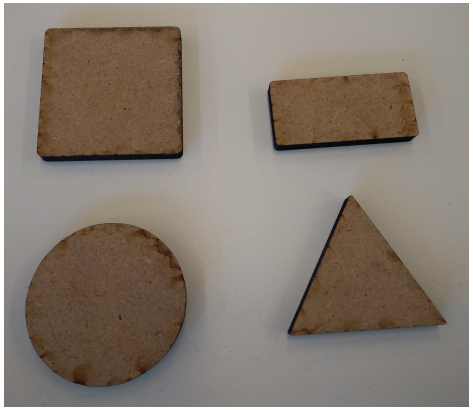


Tabletop: 98.7%



TouchTokens…

…are nice but the initial idea was even nicer!

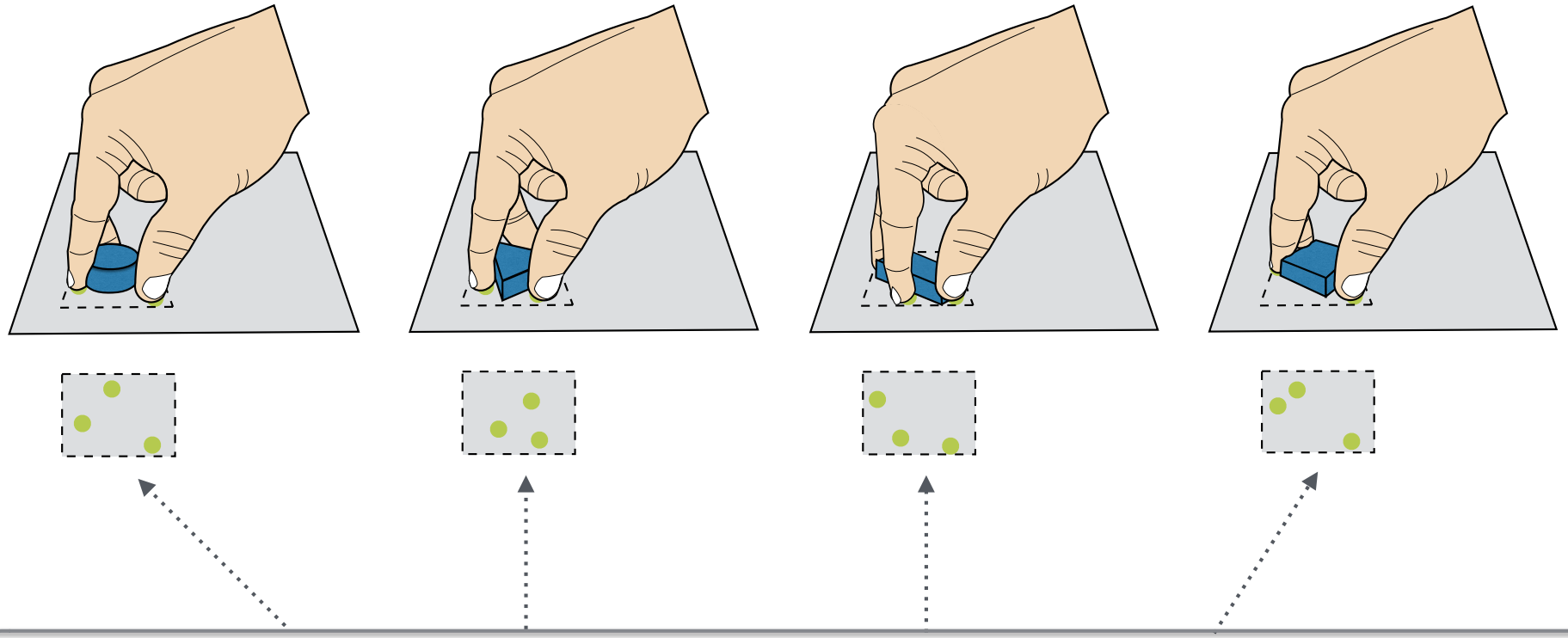


Token without notches

Everyday objects

Let’s see where we failed

Initial hypothesis

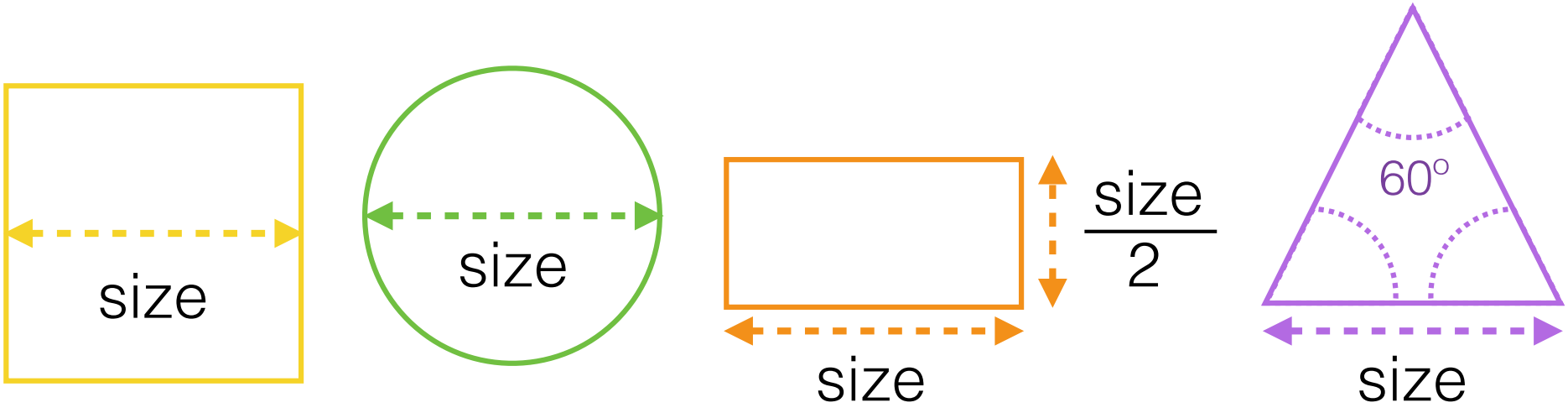


The shape of an object suggests a specific grasp, and thus a specific touch pattern. Objects that significantly differ in shape could thus be discriminated based on their touch pattern.

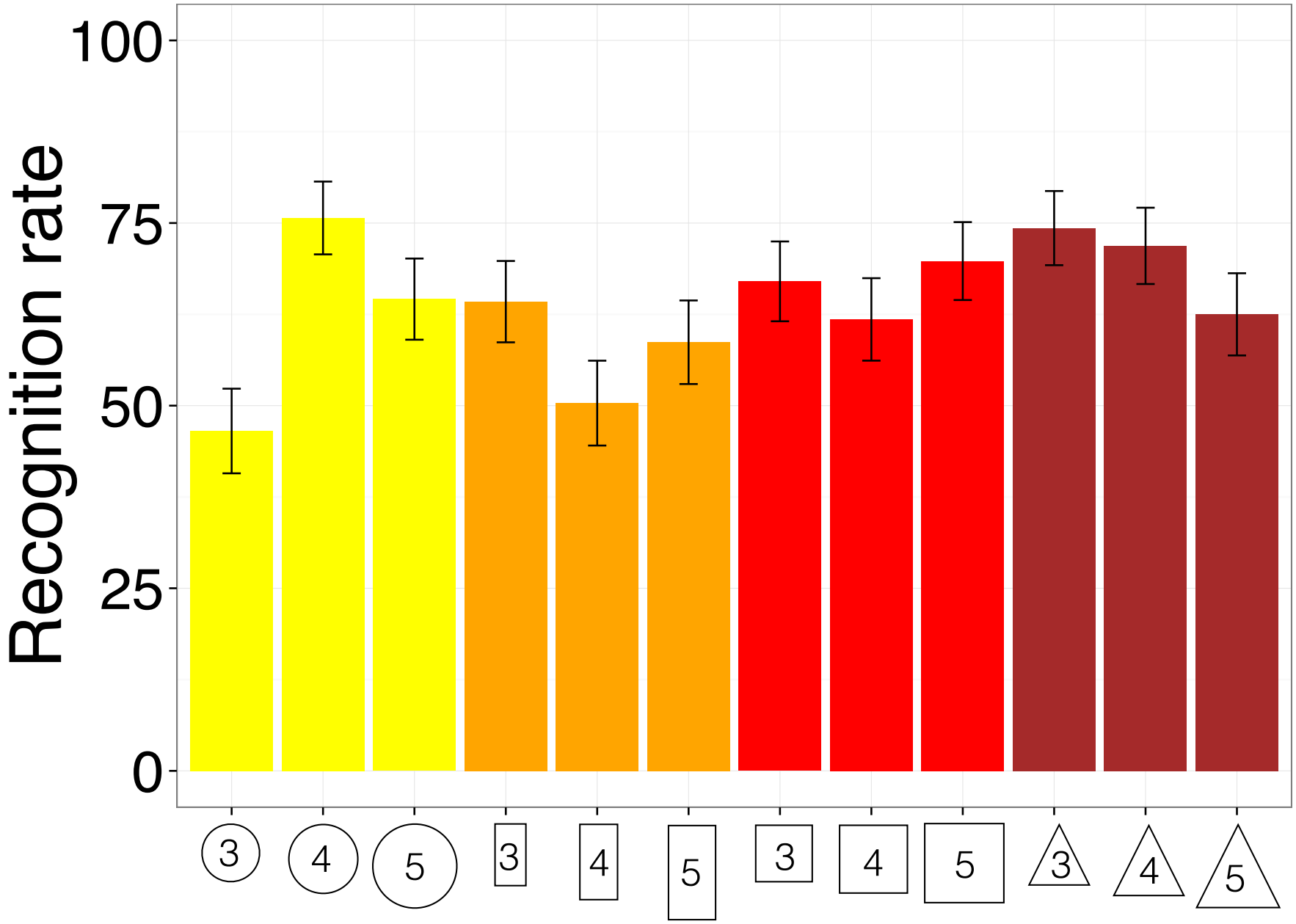
First study

Collection of touch patterns
12 “regular” tokens (no notches)

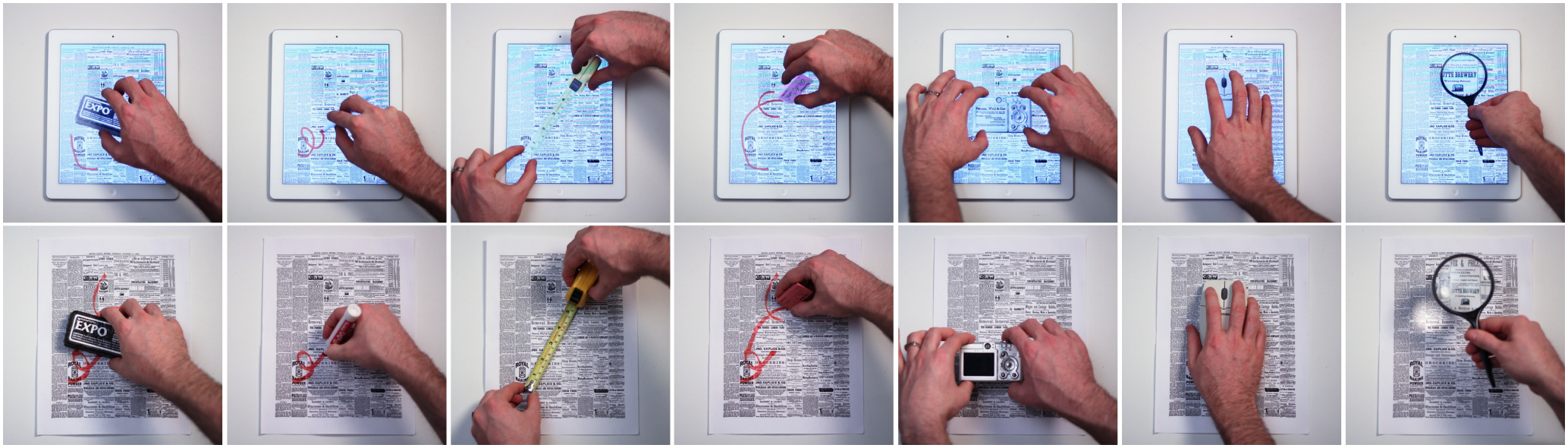
- Size = {3cm, 4cm, 5cm}
- Shape = {Square, Circle, Rectangle, Triangle}
- Size x Shape = 12 geometries



Results



Could ML help?



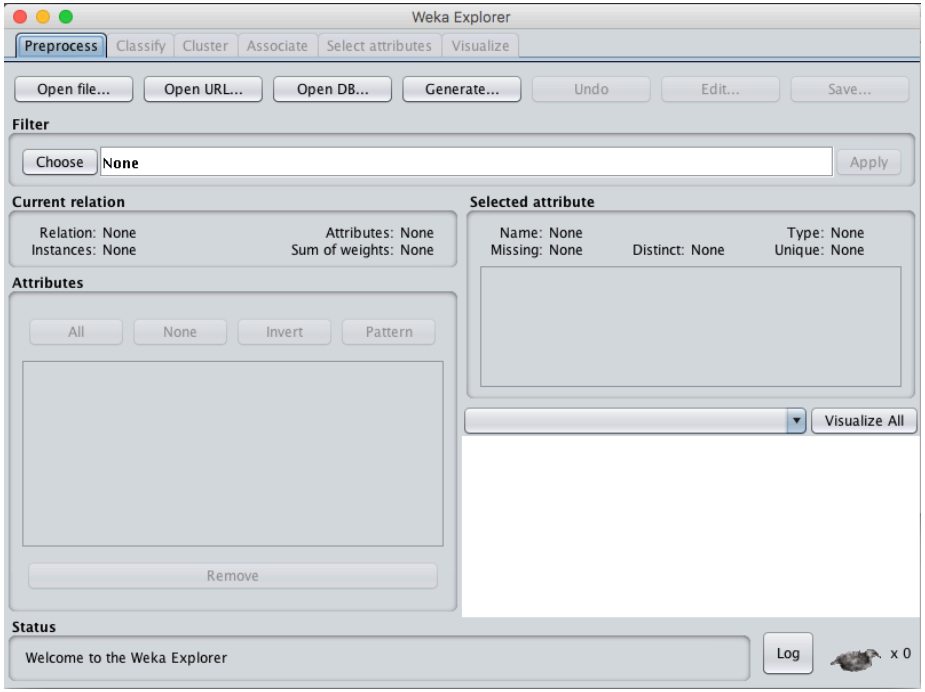
TouchTools

Chris Harrison, Robert Xiao, Julia Schwarz, and Scott E. Hudson. CHI '14. TouchTools: leveraging familiarity and skill with physical tools to augment touch interaction.

147 training instances in total (7 participants x 7 tools x 3 repetitions)
[…]
Our **features** are as follows: the number of touch points, the estimated total touch area, […], angles between consecutively-clockwise points as measured from the centroid, and the size of each point. These features are then fed into a **support vector machine classifier** trained on previously recorded data.

Let’s give it a try

scikit-learn? Weka?



Weka explorer: a GUI! Nice!

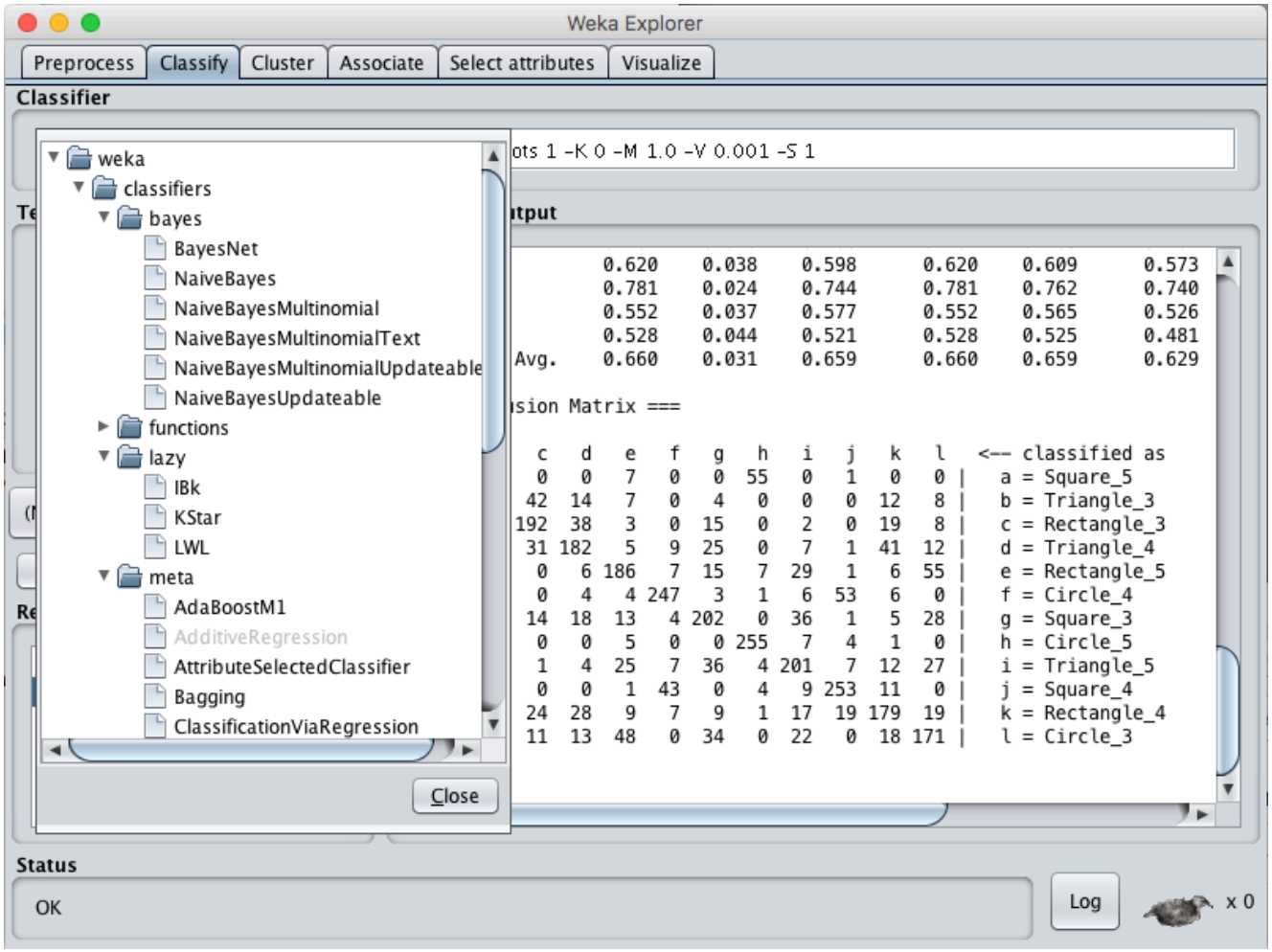
Feed Weka explorer

What features?
Let's try all the features we can think of

features.arff

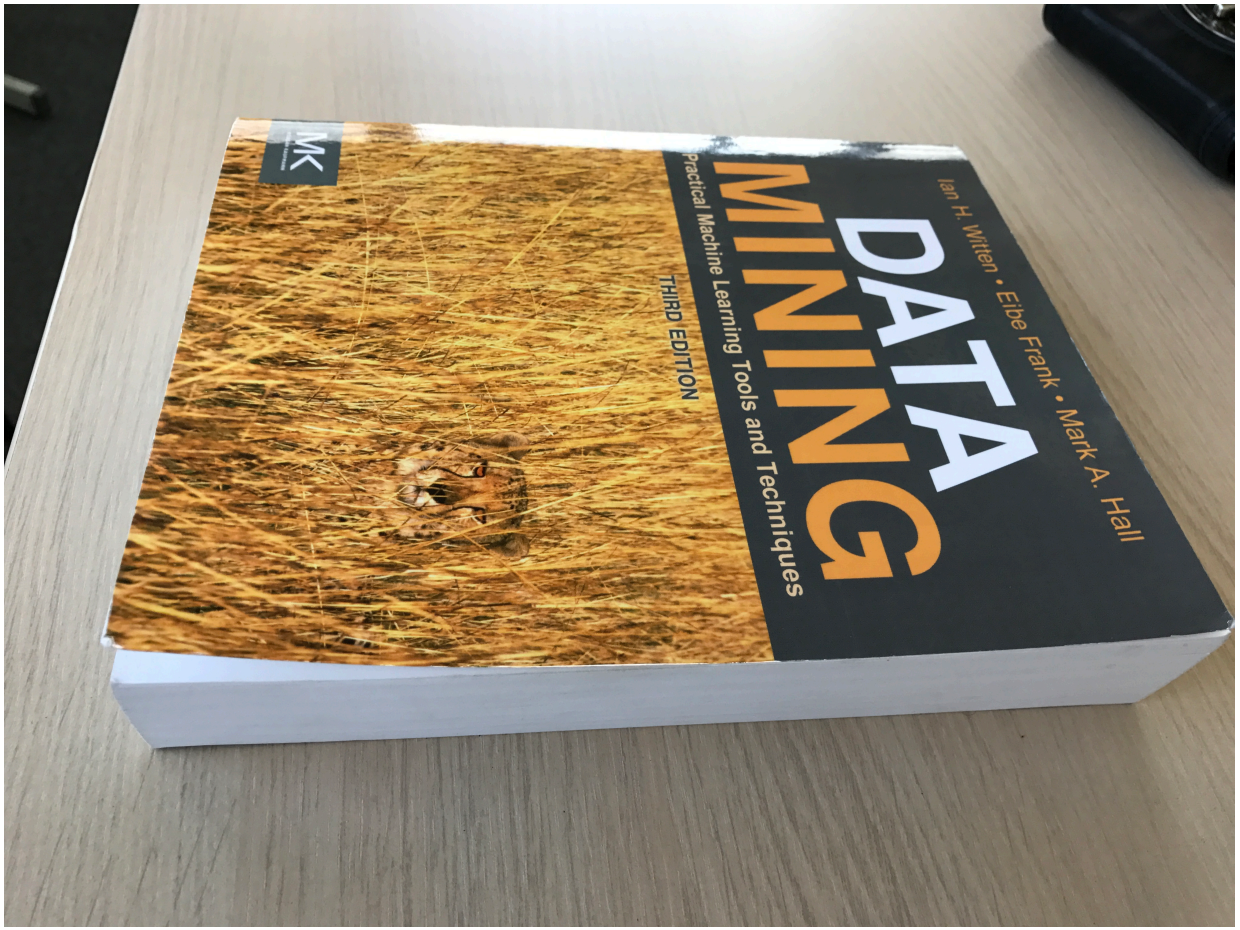
```
@relation whatever

@attribute Participant numeric
@attribute FingersCount numeric
@attribute TouchArea numeric
@attribute SmallestInscribedCircleRadius numeric
@attribute MeanPairedDistance numeric
@attribute MedianPairedDistance numeric
@attribute MinPairedDistance numeric
@attribute MaxPairedDistance numeric
[...]

@data
2, 4, -9850, 80.42, 120.68, 127.97, 46.69, 163.67, 43.31, 75.95, 69.7, 60.37, 104.03,
19.29, 81.17, 80.42, 80.42, 83.43, 1.5, 1.57, 1.67, 0.7, 2.24, 0.64, Square_5
2, 3, -1605, 36.8, 62.34, 64.4, 51.79, 70.84, 9.69, 36.02, 35.36, 30.99, 41.7, 5.39,
36.8, 36.8, 36.8, 36.8, 0, 2.09, 2.17, 1.79, 2.33, 0.28, Triangle_3
```

Explore with Weka Explorer



So many concepts to understand…

Learn Machine Learning



It takes (way) too much time…

In the end…

Alternative approaches we considered led to significantly poorer performance. In particular, we tested k-Nearest-Neighbour (k=1 and k=3) and SVM algorithms, using both raw data and describing features. The raw data was pre-processed to make it independent from rotation angle and finger identification. The describing features we considered included the touch envelope's area, as well as various descriptive statistics (min, max, mean, median and standard deviation) for measures such as point-centroid distance, distance between successive points, distance between any pair of points, etc. **These machine learning approaches yielded recognition rates ranging from 50% to 85% per participant**.

R. Morales Gonzalez, C. Appert, G. Bailly and E. Pietriga. CHI '16.
TouchTokens: Guiding Multi-Touch Patterns with Passive Tokens.

Why?

Is ML not able to help?

- Is the problem unsolvable?
- Is the problem not captured in the right way (i.e., wrong describing features)?

Did we just fail to use it properly?

- The design space for ML-based recognizers is very large: type of classifier, feature selection, size of training set, …

Why?

Is ML not able to help?

Is the problem unsolvable?

Is the problem not captured in the right way (i.e. wrong describing features)?

We could probably answer this with good tools

Did we just fail to use it properly?

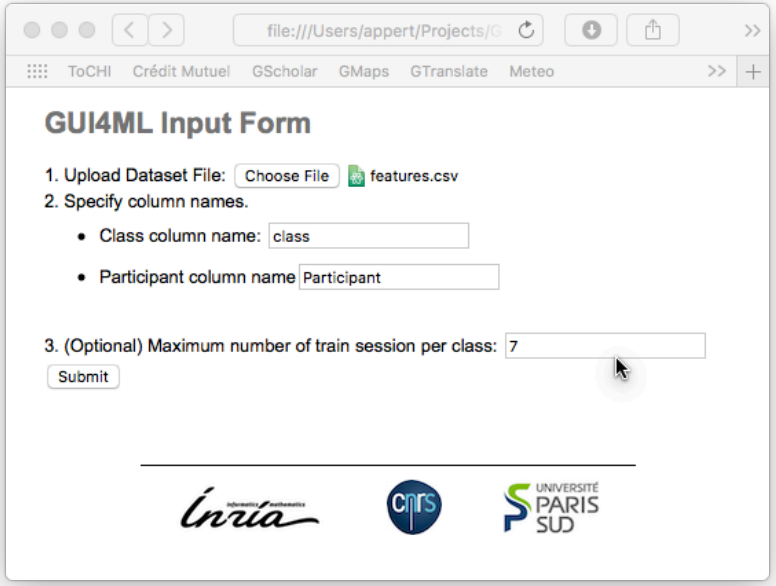
The design space for ML-based recognizers is very large: type of classifier, feature selection, size of training set, …

What would be nice for designing MT recognizers

Collect empirical data

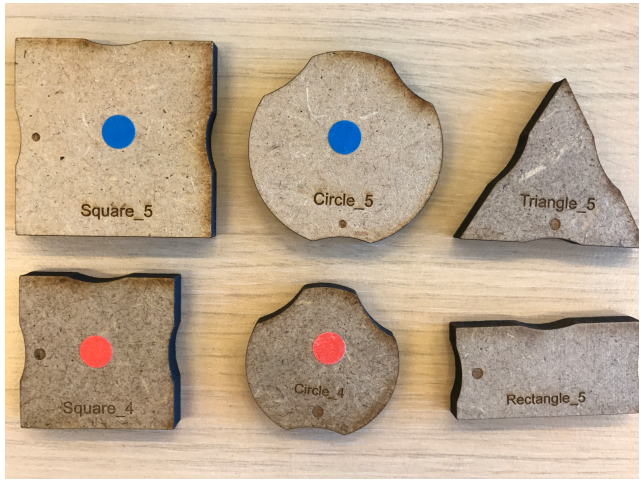
Compute describing features

Load it in a very simple GUI…



…that <u>systematically explores</u> the design space of ML-based recognizers and tells me what the best solution is with <u>different training strategies</u> (global vs per-user)

Multi-touch recognizer and training



Pre-defined interfaces should ideally rely on a global training strategy

train with *n* users and test with the $(n+1)^{th}$ user

Customizable interfaces require a per-user training strategy

train with (x - y) examples from user *n* and test with the y remaining examples from user *n*

Multi-Touch & Machine Learning

Take-away message

We need easy-to-use tools that systematically explore the design space of ML-based recognition engines

Open question

Could ML help in designing incremental recognition engines for gestures with a continuous effect (e.g., pinch-to-zoom)?

Multi-Touch & Machine Learning

Take-away message

We need easy-to-use tools that systematically explore the design space of ML-based recognition engines

Open question

Could ML help in designing incremental recognition engines for gestures with a continuous effect (e.g., pinch-to-zoom)?