Pendulum

(or the augmented machine)

What we want to achieve

- Brain-Computer-Interface
- Using our brain to speedup machine capability in solving hard problems.

Background

- Several problems seems intuitively solvable, where as an analytic solution appear to be too complex to comprehend.

- This apply to automatically optimize a system when a manual tunning is obvious and the computation require is too intense for current state of the art computers.

- The idea is simply to use the human brain to ‘augment’ the intelligence of the machine !

My dream

- When I was a kids I always though I can devise a way to talk to a machine is a seamless way

- I was thinking that the heart beats are easily controllable by the mind and emotion (seeing how it accelerates when I was shy near a person I liked)
- Why not monitor the heart beats to extract few data we want to communicate to the machine ? (see also heartmath)

What we need

- A reliable interface between the brain and the machine

- Download link : visual signal, + audio signal are the most usable link for machine to human transfer, and the display technology is ready for that !
- Upload link : no real high bandwidth link exists today !

- Let’s make it !

What are the available channels ?

What are the available channels ?

emotions can easily be conveyed by observation of few biometrics

- Skins conductivity
- Blood composition
- Heats frequency modulation
- EEG
- Gestures

The fast way …

- Our visual cortex is intimately connected to our moto-neuros (α and γ), let’s exploit this !

- Not having any sensors but a simple webcam …
- Decided to use micro muscles movements from the γ moto-neurons
- For a proof of concept I amplified the movements with a pendulum
- And have a camera “observing” the experiment, if the computer is capable of detecting variation in the movement of the pendulum, then I have a way encoding a human communication without a need to give a conscious command to the machine
- Voilà : a low bandwidth interface from brain to machine…

Test Data

- One bit link is my starting point : allow the computer to decipher a **yes** from a **no .**

pre-training

- We give the machine a large set of sample, both “yes” and “no”
- it will adjust its parameters to discriminate the two states w/o knowing which one is which, this will allow a faster learning once the feedback is provided with new samples.

- **Training set :**
- [QmVsWrszWGrCdDQ5R8BtUBX1Bg6Sn2XfWnKpUDEg3NxH1K]

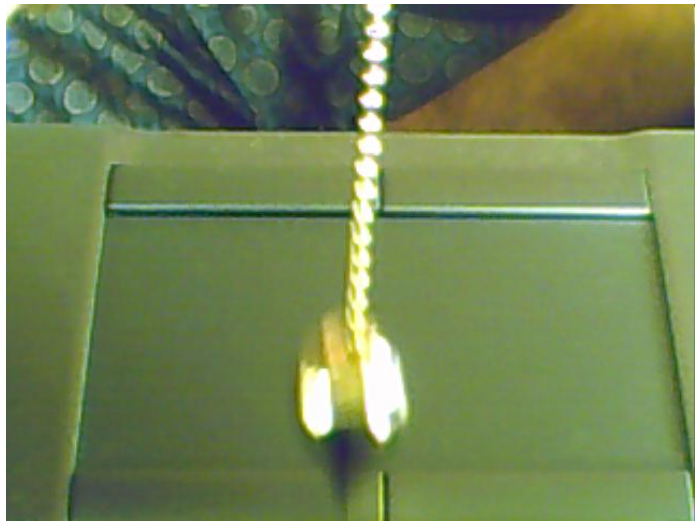
The Learning …

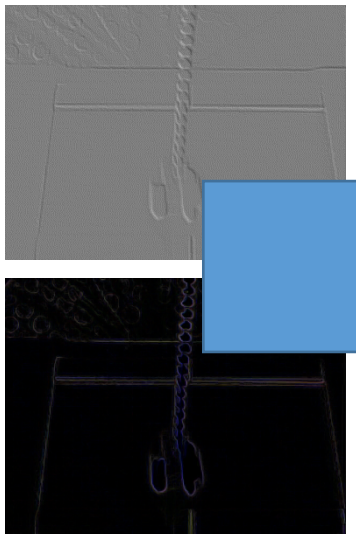
- provide the inputs stimuli : the video clip
- give immediate feedback on decision providing by the machine.

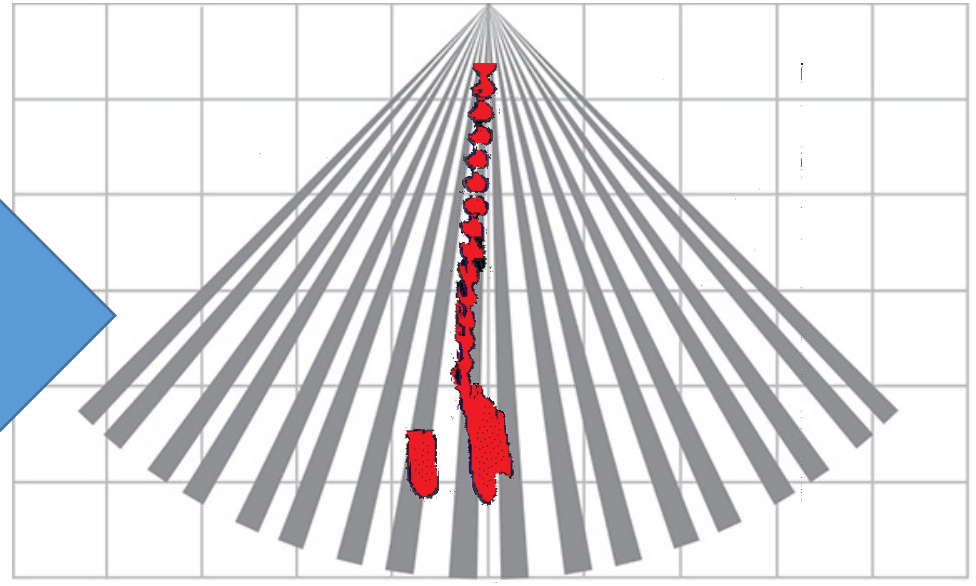
the perfect “eye”

Give the machine an “eye” for the task

- **sensitive to pendulum shape and color**
- **sensitive to motion**





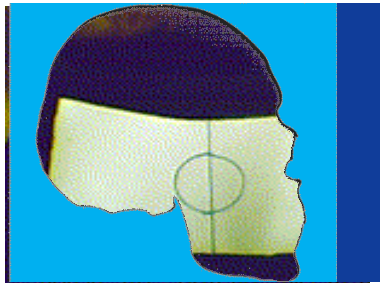


Input signal pre-formating

- **Simplify input to make learning easier**
- **Polar representation …**
- **Image segmentation, “pendulum” recognition**
- **Separate moving from background**
- **Correlate with “pendulum” model**
- **Recover hand µ-movements**
- **Conic grid filter**
- **Movement signature**

Input signal pre-formating

- **Simplify input to make learning easier**
- **Polar representation …**
- **Image segmentation, “pendulum” recognition**
- **Separate moving from background**
- **Correlate with “pendulum” model**
- **Recover hand µ-movements**
- **Conic grid filter**
- **Movement signature**



3D motion extraction …

In order to ease the learning we will teach the machine a few tricks !

1. Recognizing a pendulum
2. Tracking the pendulum centroid
3. Learning what depth is
4. Learning what motion is
5. Recognizing linear motion
6. Recognizing rotational motion

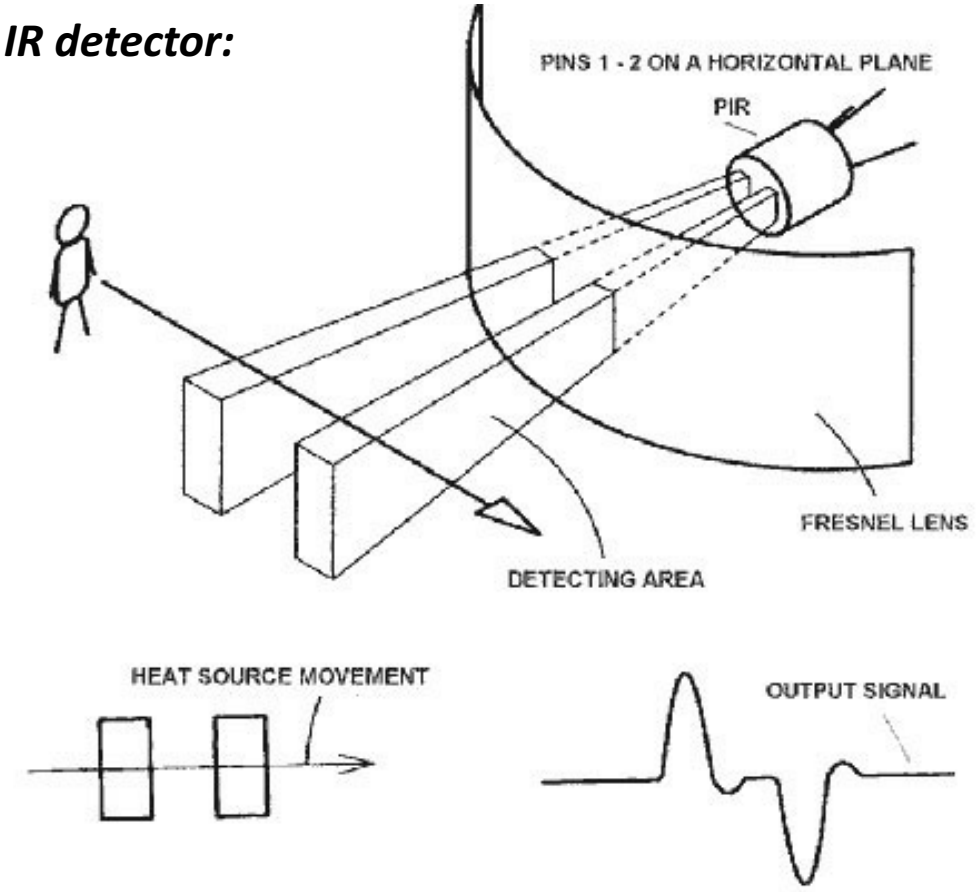
frequency domain …

Everything is easier in Fourier space …

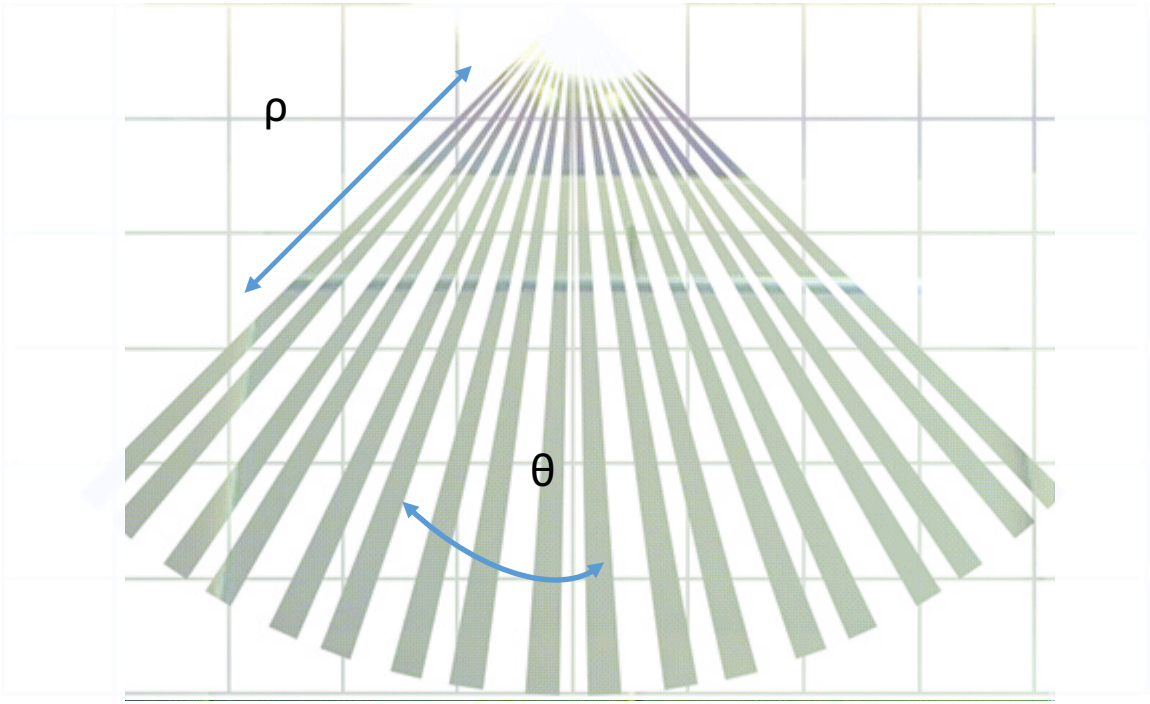
- Centroid Contour Distance Curves, Fourier descriptors …
- Invariances :
  - Scale : normalize coef. => depth extraction
  - Translation => dc = 0 => linear motion
  - Rotation => phase of coef.

View from a Fresnel lens

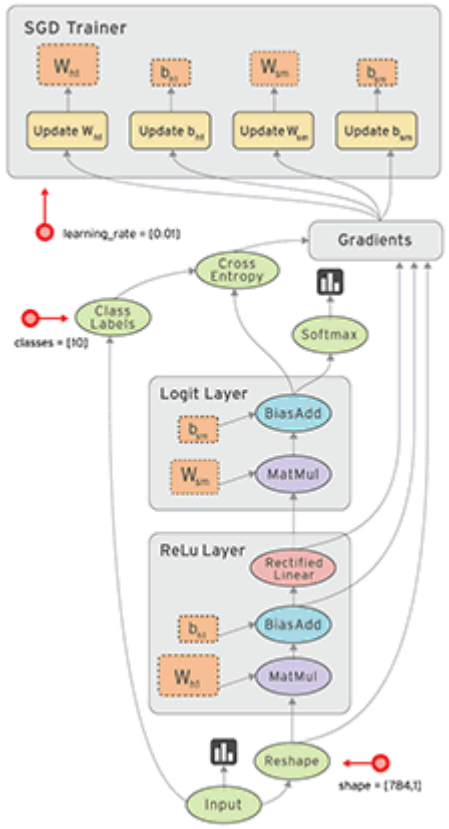
IR detector:



Pendulum motion detection



decision engine…



Multiplayer:

- 1st layer: input video
- 2nd layer:
  1. Θ channel
  2. ρ channel

Outputs:

1 bit: 0/1, yes/no