# TRA-KINESICS

Controlling screen pointer using hand gestures

### **Team Members**

Gaurav Mittal (2012CSB1013)

Kaushal Yagnik (2012CSB1039)

Anshuman Yadav (2012CSB1047)

Riya Garg (2012CSB1031)

Email: gauravmi@iitrpr.ac.in

Email: yagnikkb@iitrpr.ac.in

Email: anshumany@iitrpr.ac.in

Email: riyagarg@iitrpr.ac.in

# Acknowledgement

We are immensely thankful to our course instructor, Dr. Deepti Bathula, for her guidance throughout the project. Also, we would like to mention the websites stackoverflow.com and mathworks.com which also helped us in solving many of the trivial problems we faced.

### **Problem statement**

Since the past decade, the most common form of input peripherals used to control the pointer on the screen have been the conventional mouse, trackpad (mainly in laptops) and the more recent and less common ones being light pen and touch pad.

However, there are scenarios where using any of these devices may not be very ergonomic or convenient to use, such as during presentations or seminars. In such scenarios, the most intuitive and effective means of communication is through gestures and natural body language.

## **Solution Proposed**

We propose to manoeuvre the pointer on the screen simply using hand gestures. This serves to be optimal considering the following scenarios:

- 1. It is an excellent, ergonomic and inexpensive alternative to the already available means to control the pointer. It makes use of a webcam (which has become quite ubiquitous nowadays) with a decent resolution enough to serve the purpose.
- 2. Hand gestures are far more intuitive compared to using the mouse. They can be used to control the pointer from a distance without any physical contact with the pointing device, which can be really helpful in presentations, seminars, meetings or conferences.

### **Design Decisions**

#### 1. What to detect?

- Gestures Recognitions through fingers ( hand )
  - Fingers seems to be the natural way to control the mouse pointer
  - Also with less efforts the fingers can be used to create different gestures.

#### 2. How to detect hand gestures?

- Use Machine Learning
  - Training data set (image of hand gestures)
  - Test image, classifying the image gestures
    - Advantages:
      - Fast processing on the test data.
    - Difficulties:
      - Large training time.
      - Need to be exhaustive
      - Adding a new gesture would need re-learning the model
      - Generalization issues

#### > Pre-processing the input frames

- Noise filtration
- Template matching
  - Advantages:
    - No generalizations issues
    - Easy implementation
    - No training or learning period needed
    - Can be extended for new gestures
  - Difficulties:
    - Real time
    - Noisy inputs
    - Accuracy
  - Noise filter approaches
    - The image after the skin detection had a lot of noise from surrounding
    - The regions of the surrounding that felt in the range of skin colour were there, but however not dominant

#### > Noise filter

- Gaussian
- ❖ Median

Which one to pick?

• Since we mainly encountered salt-pepper noise in the image, we preferred to use the median filter.

#### > Image construction and enhancement using inbuilt functions?

- Clustering for the re-construction of any broken parts in the palm.
- Inbuilt functions are used for Real time processing to be as optimized as possible.

# **Technical Requirements**

- Windows operating system
- Webcam with 'YUY2\_320x240' support
- Matlab Compiler

### **Implementations**

#### MyGauss()

The function MyGauss() returns a Gaussian 2-D filter with given mean, variance and size. The function is uses the property of Gaussian function to generate the filter.

#### MyConv(filter, image)

The function performs the convolution of the filter with the input image and returns the convolved image.

#### Detect(image)

The function performs a template matching on the processed image that only consists of the palm. The function uses MyGauss, MyConv and the difference of Gaussian approach to create a template of the finger. The function returns the response of the image to the template as its output.

#### MouseGoTo()

Move to mouse pointer to the desired location using java class Robot.

#### MouseMove()

Move the mouse pointer provided the difference in the coordinates using java classes Robot.

#### MouseLeftClick()

Left Click the mouse pointer at the given location using java classes Robot and InputEvent.

#### MouseRightClick()

Right Click the mouse pointer at the given location using java classes Robot and InputEvent.

# **Primary Challenge Faced**

\* Real time processing of the video from the webcam without introducing a lag.

# Current limitations of the project

- ❖ Not invariant to occlusions.
- Partially dependent on the lighting conditions.
- Not invariant to objects with colour same as that of skin.
- ❖ Resolution of the webcam limited to 320x240 to avoid lags in the real time processing.

### Proposed improvisations and extensions

- Control gestures limited to left-click, double-click, moving, and dragging the pointer. We can add additional gestures like right-click, scrolling up and down etc. for a better control.
- Use powerful multi-core processors to enable high resolution video processing from the webcam. This allows controlling the pointer more smoothly compared to a lower resolution.

### References

- http://www.sciencedirect.com/science/article/pii/S0262885603000702
- http://www.ijarcce.com/upload/2013/november/48-a-Albert prasanna kumar simulation.pdf
- http://www.academia.edu/5426093/Gesture Recognition by Neural Networks using MATLAB
- https://en.wikipedia.org/wiki/Gesture recognition
- http://in.mathworks.com/help/