

# JANUS

View Dependent Desktop VR using Head Tracking



# THE IDEA

- Creating a realistic 3D environment and virtual reality only with help of a webcam and computer screen
- Impression of threedimensionality is achieved by altering the screen content according to the viewpoint of the user
- The challenge is to estimate the viewpoint

# REQUIREMENTS

- Algorithm for identifying the user's face in front of the screen
- Method for shifting and scaling an image of a landscape based on the user's heads's movement (virtual window)
- Future Work: Providing a OpenGL-rendered 3D world reacting to the user's movements

# FACE DETECTION - PING PONG BALL

- First Approach: Cap with orange colored ball
- Searching for points having a predefined threshold of hue, saturation and intensity
- Median over these points as x- and y-, count as z-coordinate



# FACE DETECTION - PING PONG BALL

- Result was fast and quite stable
- Several disadvantages:
  - User has to wear a cap
  - hue, saturation and intensity highly depend on lighting conditions
  - no other orange objects in the camera's field of view permitted
- ➔ Approach was quickly abandoned

# FACE DETECTION - CLASSIFIER CASCADES

- Using OpenCV's built-in head tracking function,  
`cvHaarDetectObject`
- Loaded with a classifier cascade, it finds the region most likely to contain the object specified in the cascade



# FACE DETECTION - CLASSIFIER CASCADES

- Poor results (low speed, inaccurate) after applying to the original image
- Tweaking the performance included
  - Scaling down the input image
  - Histogram equalization
  - Switching to a newer OpenCV version (1.1.0), which supported the `CV_FIND_BIGGEST_OBJECT`-flag
  - Taking the average over some of the last found points to iron out shaking

# FACE DETECTION - TRACKING POINTS

- Identifying the user's actual viewing direction
- Finding a few popular, good-to-track- points
- Next Step: Calculating the position matrix of the head, again using an OpenCV function



# SPEED

- Head tracking in its current form can be done in real-time
- However, performance can be further optimized by separating the head tracking and drawing functions into different threads
- Drawing interpolated points to compensate slow tracking

# QUESTIONS ?

Visit our homepage:

<http://home.iitb.ac.in/~08305204/dip/janus>

