Multi-Touch- and Multi-User-Interfaces
Björn Frömmer
Goals of the first lab assignment:

- Part 1: Blob Detection utilizing GIMP and two screenshots from the multi-touch system (images will be provided)

- Part 2: Blob Detection using openCV and a videostream from the multi-touch camera (video will be provided. Do NOT use the still images for this part!)

  - Find parameters that make sense on your own (dynamical manipulation of these parameters possible through keyboard-input during run-time):

    ```cpp
    if( waitKey(1) == 27 ) // "esc" key pressed?
    {
        std::cout << "EXITING: User stopped the process.\n\n"
        break;
    }
    ```
Lab assignment 1, Part 1

• **Image processing (GIMP)**
  • Background Subtraction (difference image)
  • Highpass filter (blur + difference image)
  • Segmentation (binarization, threshold)
  • Edge detection (~Sobel)
Camera image, empty
Camera image, hand visible
Difference image (background sub)
Difference image

- Brightness and contrast amplified (only for the slides so that you can see it)
Binarization alone is not sufficient to segment the fingers!
Highpass + Binarization

Now it works!
Positions, IDs

id = 1

id = 2

id = 3

id = 4

id = 5
Lab assignment 1, Part 2

- **video processing (OpenCV)**
  - Background subtraction (difference image)
  - Segmentation (highpass*, binarization)
  - Edge detection (find contours)
  - Object recognition (position, size)

*see on next slide*
// background subtraction
absdiff(...);

// simple highpass filter
blur(...);
absdiff(...);
blur(...); // optional

// segmentation (binarization)
threshold(...);

// find contours
findContours(...);
findContours()

vector<vector<Point>> contours;
vector<Vec4i> hierarchy;

findContours(binary, contours, hierarchy, CV_RETR_CCOMP, CV_CHAIN_APPROX_SIMPLE);

// iterate through all the top-level contours -> "hierarchy" may not be empty!
if(hierarchy.size() > 0)
{
    for(int idx = 0; idx >= 0; idx = hierarchy[idx][0])
    {
        // check contour size (number of points) and area ("blob" size)
        if (contourArea(Mat(contours.at(idx))) > 30 && contours.at(idx).size() > 4)
        {
            ellipse(original, fitEllipse(Mat(contours.at(idx))),
                    Scalar(0,0,255), 1, 8); // fit & draw ellipse to contour at index

            drawContours(original, contours, idx, Scalar(255,0,0), 1, 8,
                         hierarchy); // draw contour at index
        }
    }
}
What comes next?

- Blob tracking (nearest neighbor algorithm)
- Interface (create events + send them)
- Client application (create a “proof of concept” application that receives data from the tracker)