Multi-Touch- und Multi-User-Interfaces

Björn Frömmer

Leitfaden

Ziele des ersten Praktikums:

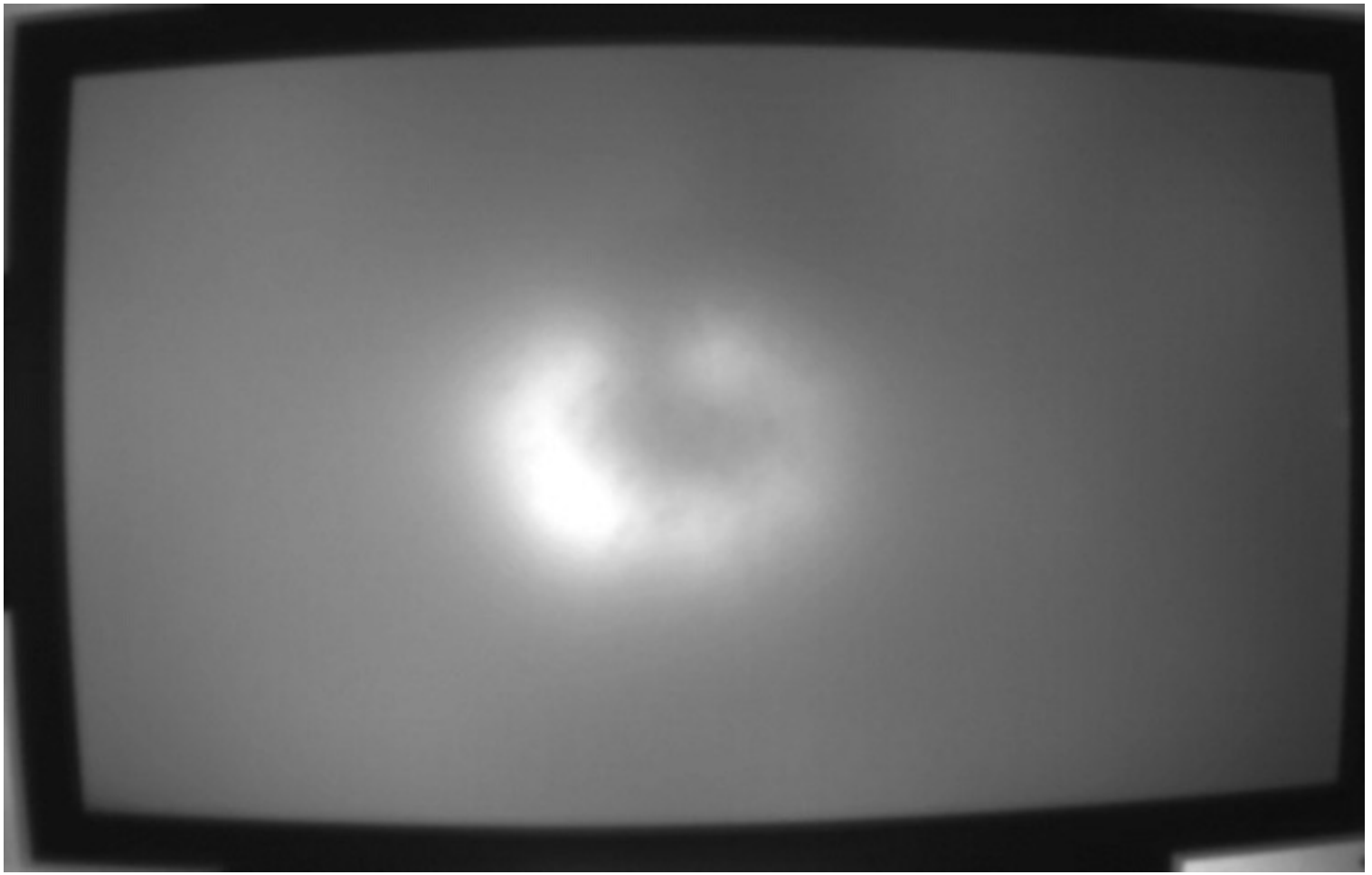
- Teil 1: Blob Detection mit GIMP und zwei Einzelbildern der Multitouch Kamera (Bilder werden bereit gestellt)
- Teil 2: Blob Detection mit openCV und Videostream der Multitouch Kamera (Video wird bereit gestellt, NICHT die Einzelbilder verwenden)
  - Finden Sie selbst geeignete Parameter für die verwendeten Bildverarbeitungsfilter! (Dynamische Anpassung der Parameter durch Tastatureingaben zur Laufzeit möglich):

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

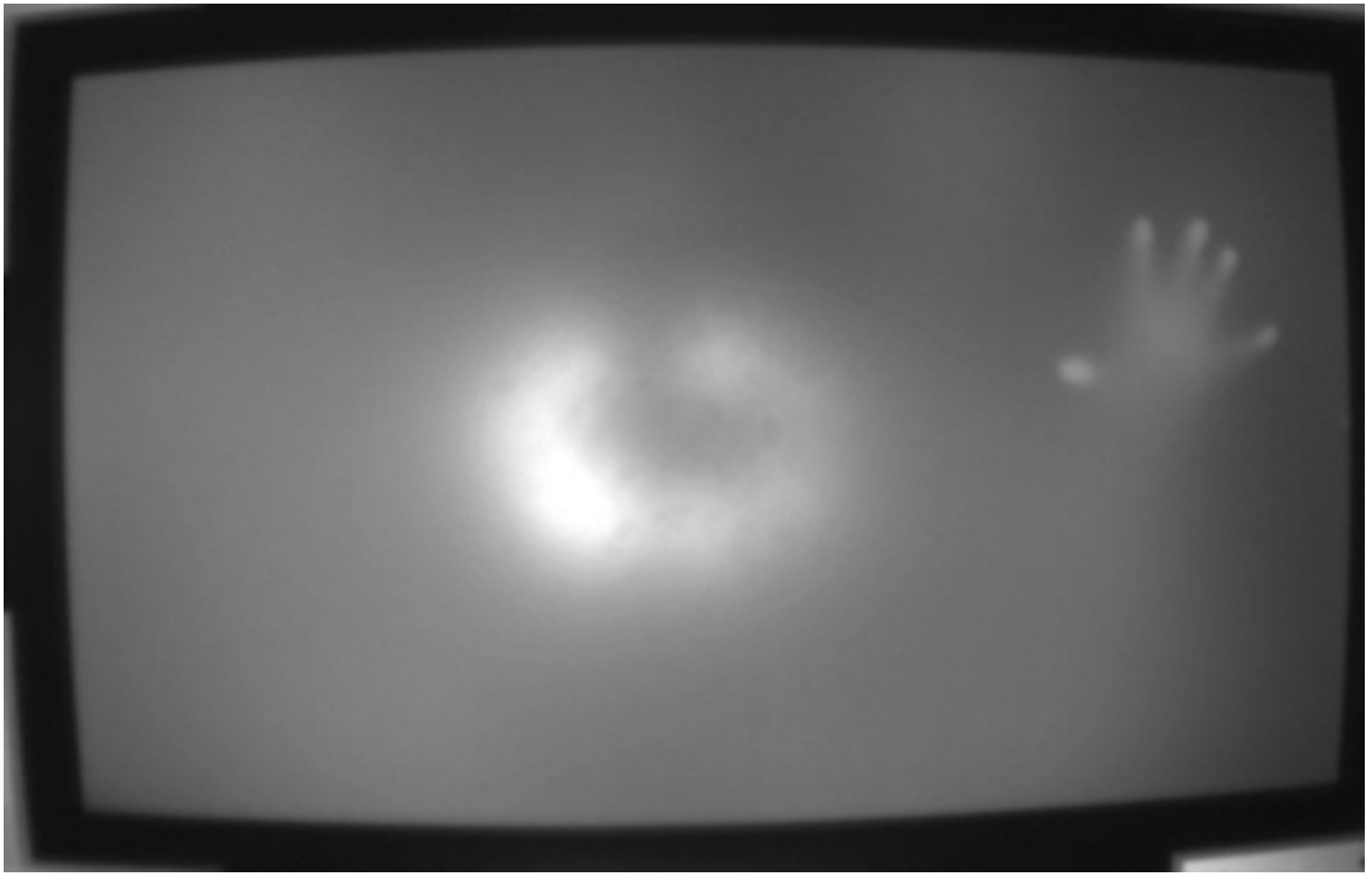
Praktikum 1, Teil 1

- **Bildverarbeitung (GIMP)**
  - Background Subtraction (Unterschied)
  - Hochpass Filter (Weichzeichner + Differenzbild)
  - Segmentierung (Schwellwert)
  - Kantendetektion (~Sobel)

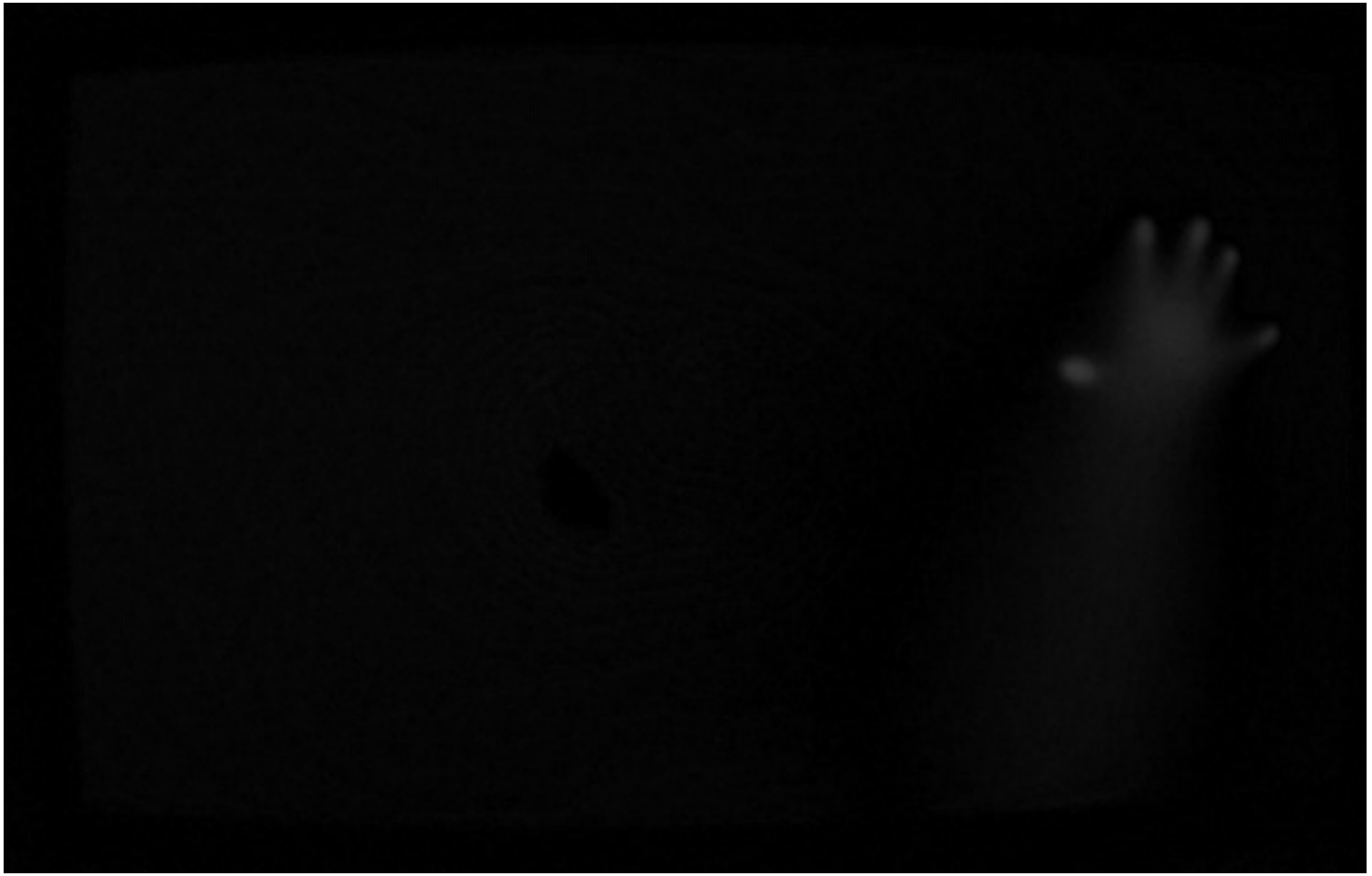
Kamerabild, leer



Kamerabild, Hand

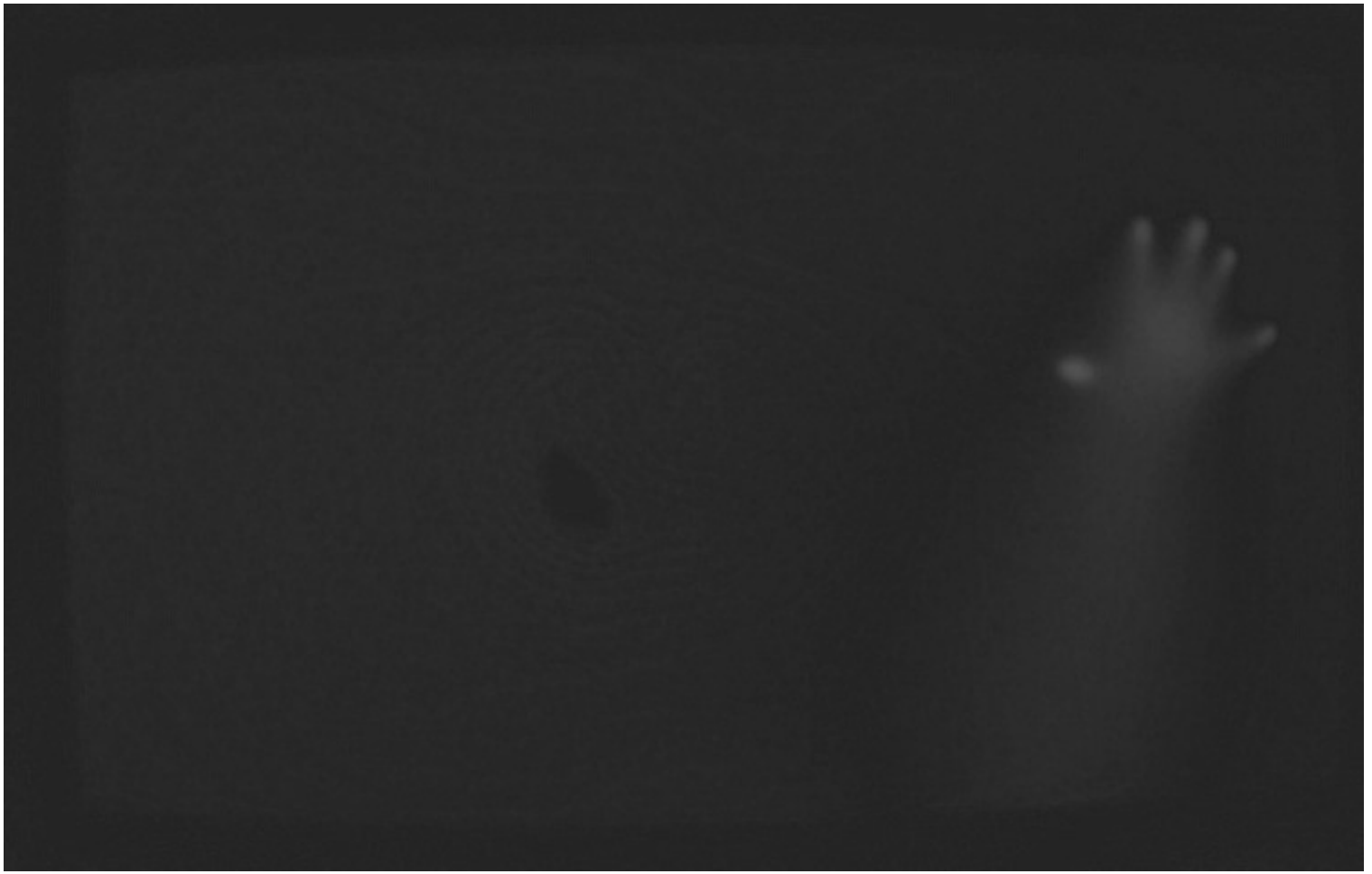


Differenzbild (Background Sub)

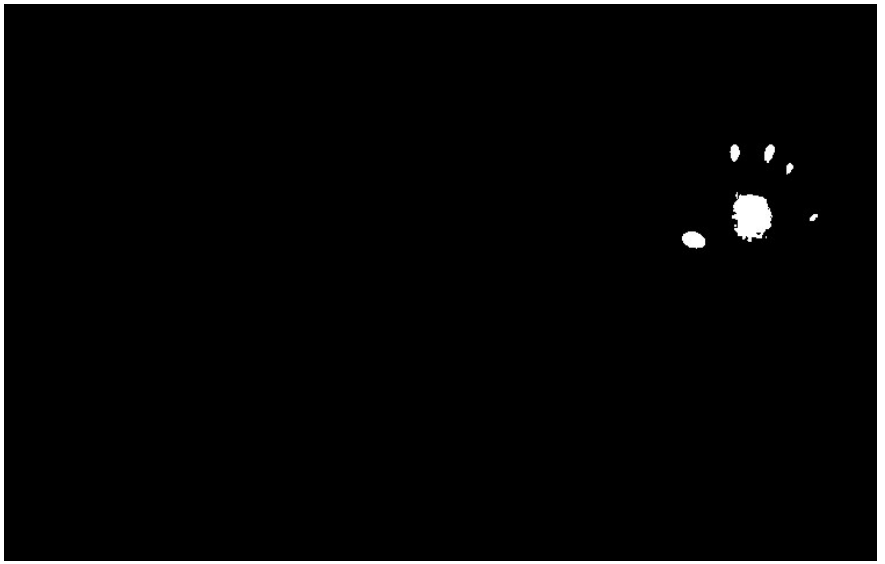


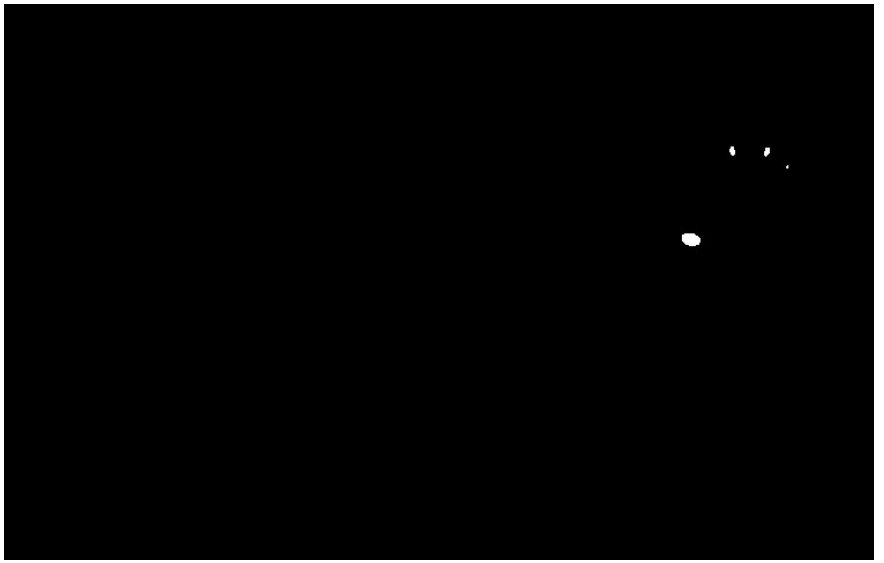
Differenzbild

- Aufgehellt und Kontrast verstärkt (nur für die Folien!)

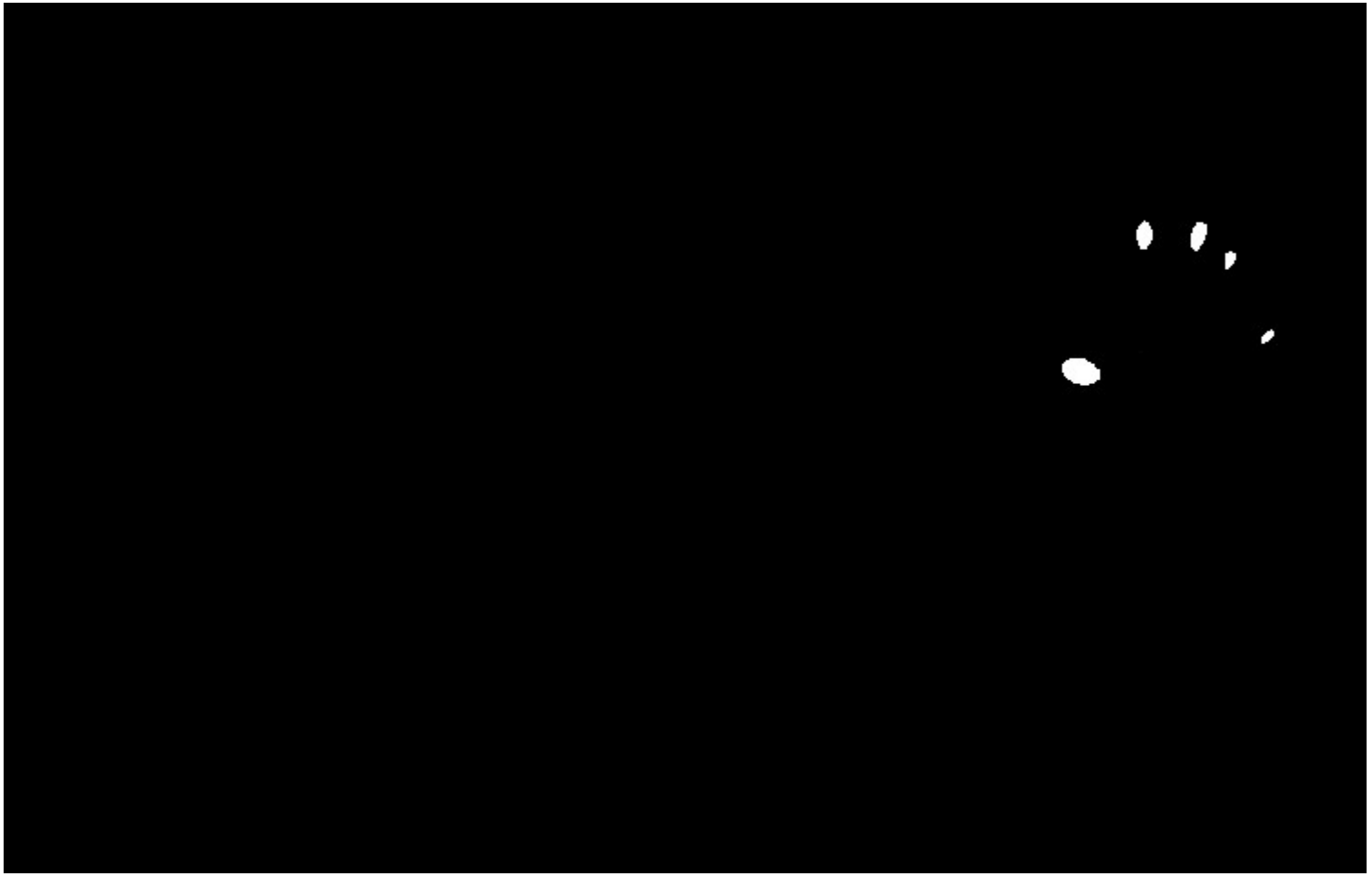


Schwellwert

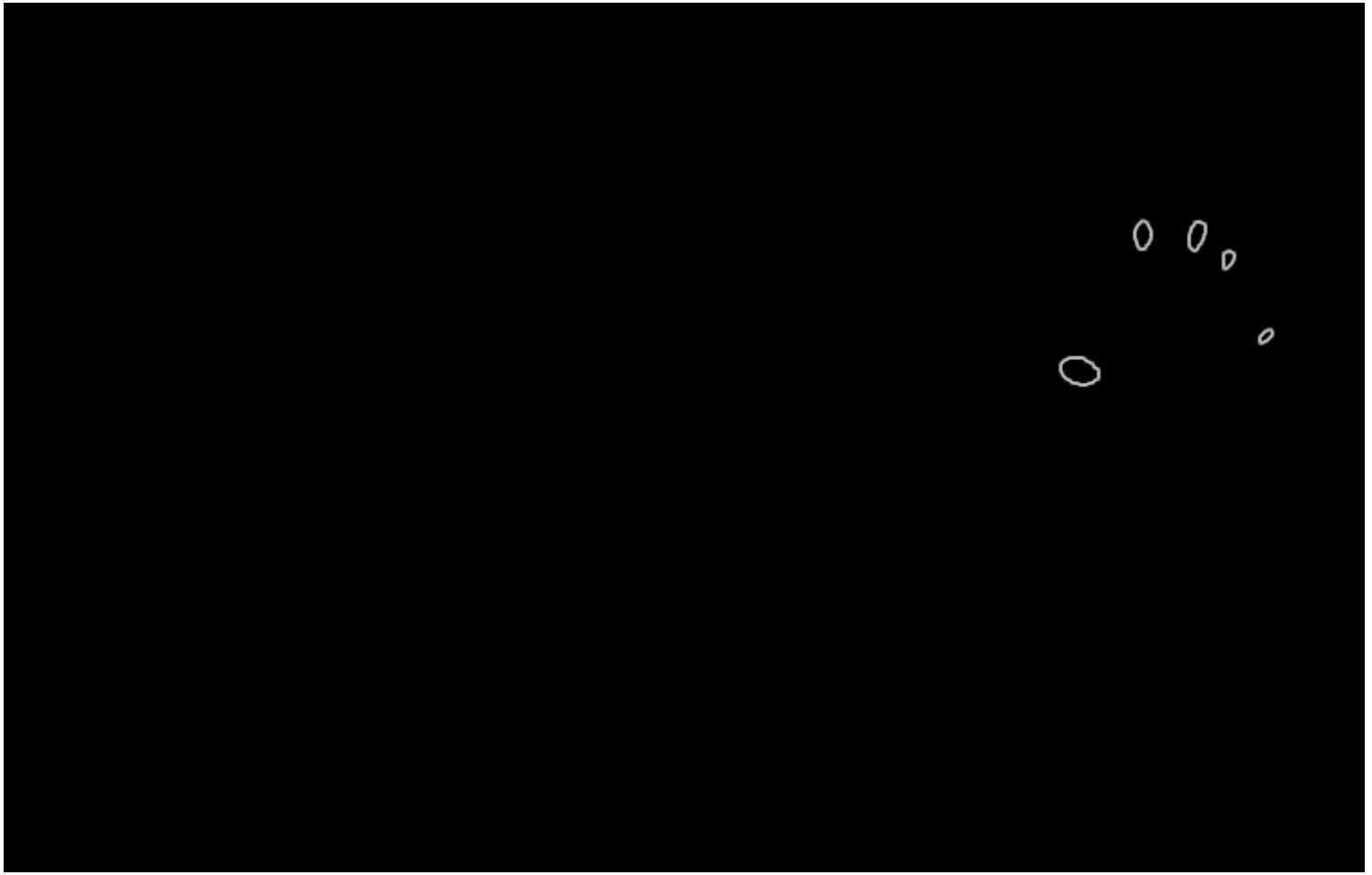




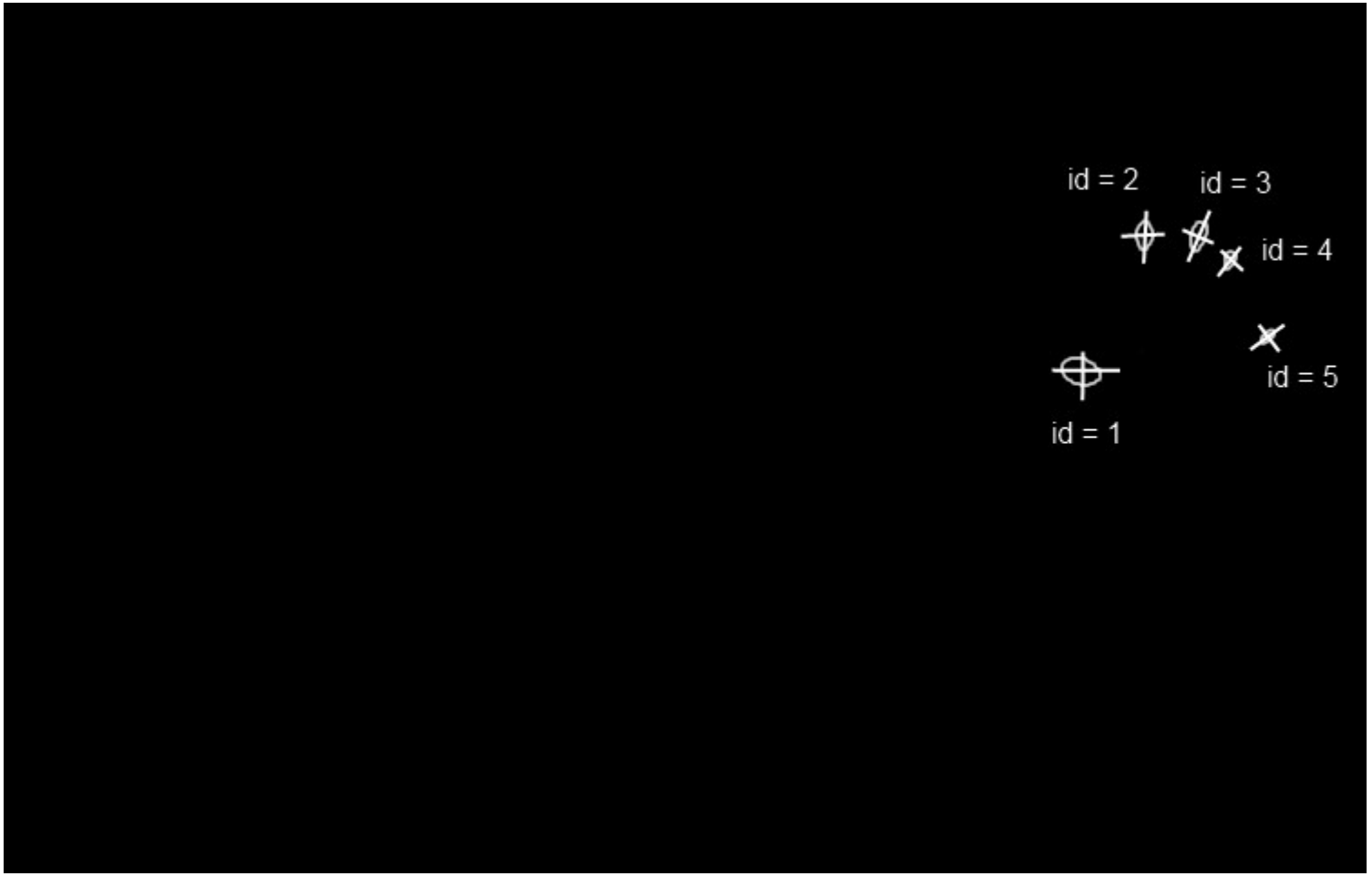
Hochpass + Schwellwert



Sobel



Positionen, IDs



Praktikum 1, Teil 2

- **Bildverarbeitung (OpenCV)**
  - Background Substraction (absdiff)
  - Segmentierung (Hochpass*, Binarisierung)
  - Kantendetektion (findContours)
  - Objekterkennung (Position, Größe)

* siehe nächste Folie

Image Filters in OpenCV

```
// background subtraction
absdiff(...);


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


// threshold
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
          }
     }
}
```

Was kommt danach?

- Blob Tracking (Nearest Neighbor Algorithm)
- Interface (Create Events + Send them out)
- Client Application (Create a simple “Proof of Concept”)