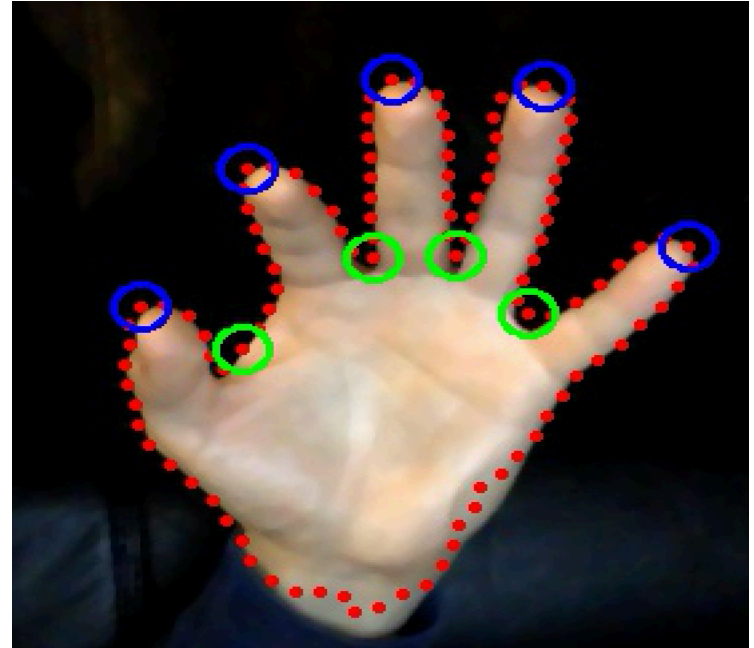


# k-Curvature Fingertip Tracking

Presented by: Yerbol Aussat  
March 2, 2018  
CS889 – University of Waterloo



# Preparatory Steps for Fingertip Tracking

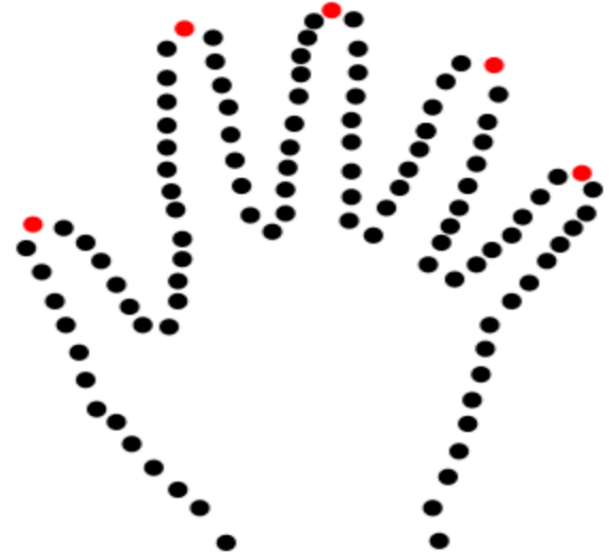
- Hand Segmentation
  - By isolating skin regions in the frame
  - Usually done in YCrCb or HSV color spaces



# Preparatory Steps for Fingertip Tracking

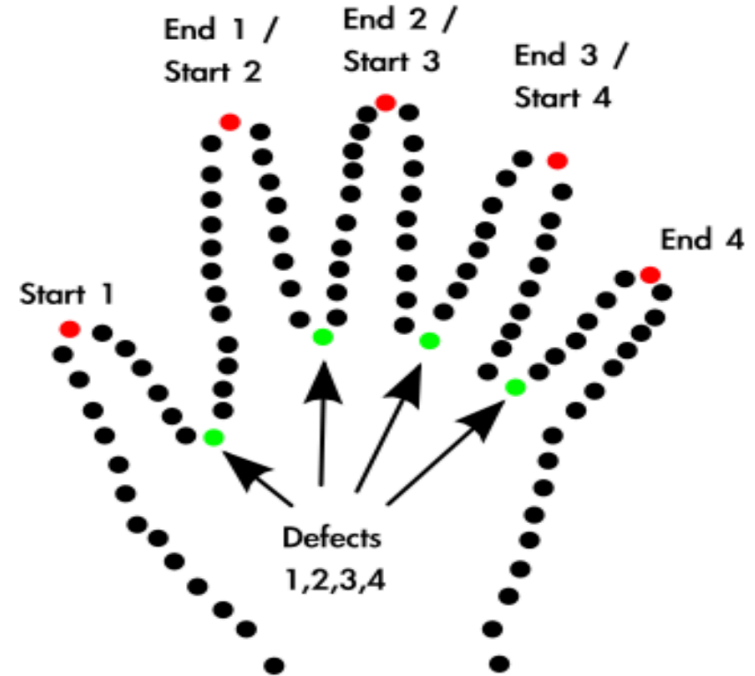
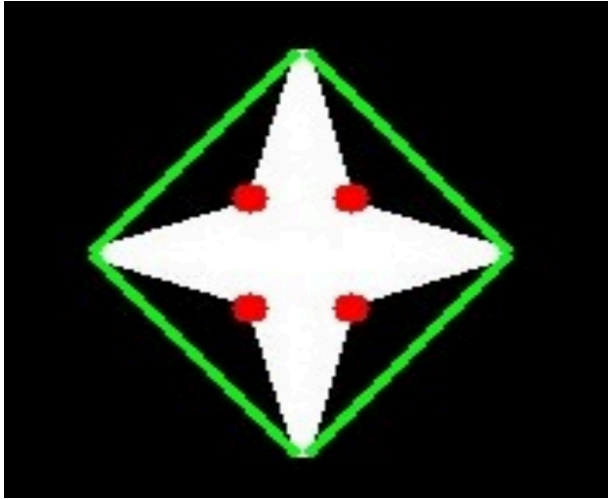
- Extracting Hand Contour
  - `cv2.findContours()` method in OpenCV
  - Returns a set of counter-clockwise perimeter coordinates

$$C_i(j) = \{(x_j, y_j)\}$$



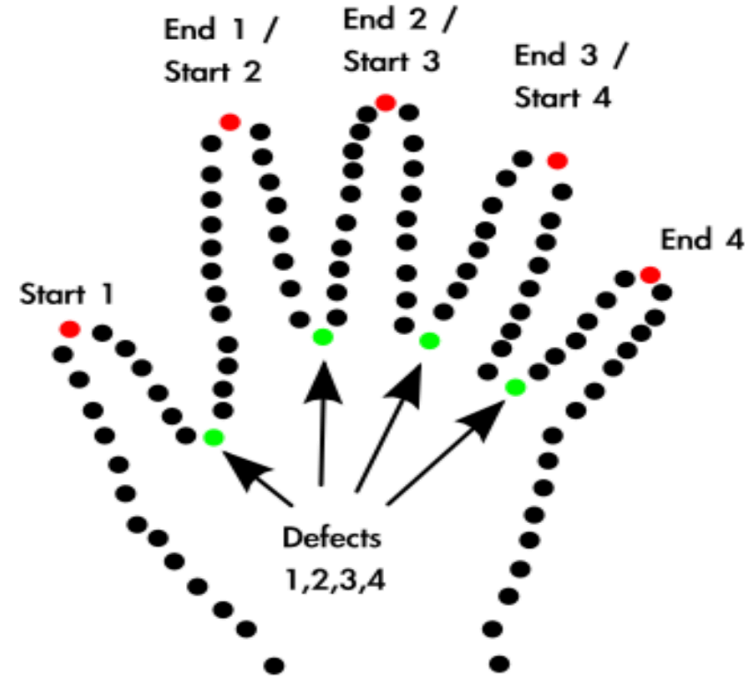
# Approach 1: Convexity Defects

- `cv2.convexityDefects()` method in OpenCV
- Identifies concave places on the contour
- Returns [ **start point, end point, farthest point, approximate distance to farthest point** ]



# Approach 1: Convexity Defects

- Using the geometry of convexity defects, the fingers can be identified.
- Thresholds can be set on:
  - Angles between two lines starting at the far point and ending at start and end points
  - Distances between far points and start / end points



# Approach 1: Convexity Defects

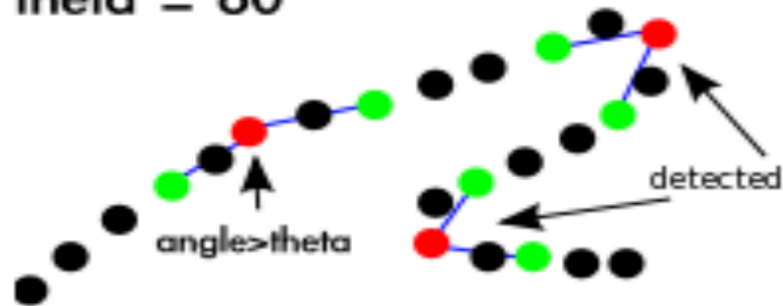
- Using the geometry of convexity defects, the fingers can be identified.
- Thresholds can be set on:
  - Angles between two lines starting at the far point and ending at start and end points
  - Distances between far points and start / end points
- **Limitation: Cannot track a hand with only one finger extended**

# Approach 2: k-Curvature

- Idea: Find pixels that represent peaks along the contour perimeters.
- At each pixel  $j$  in a hand contour, compute the k-curvature, which is the angle between the vectors  $[C_i(j), C_i(j-k)]$  and  $[C_i(j), C_i(j+k)]$  where  $k$  is a constant

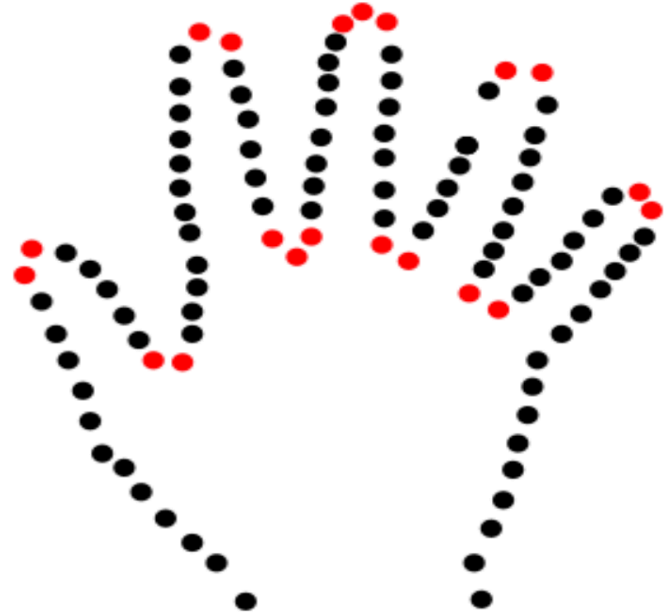
$k = 2$

$\theta = 60$



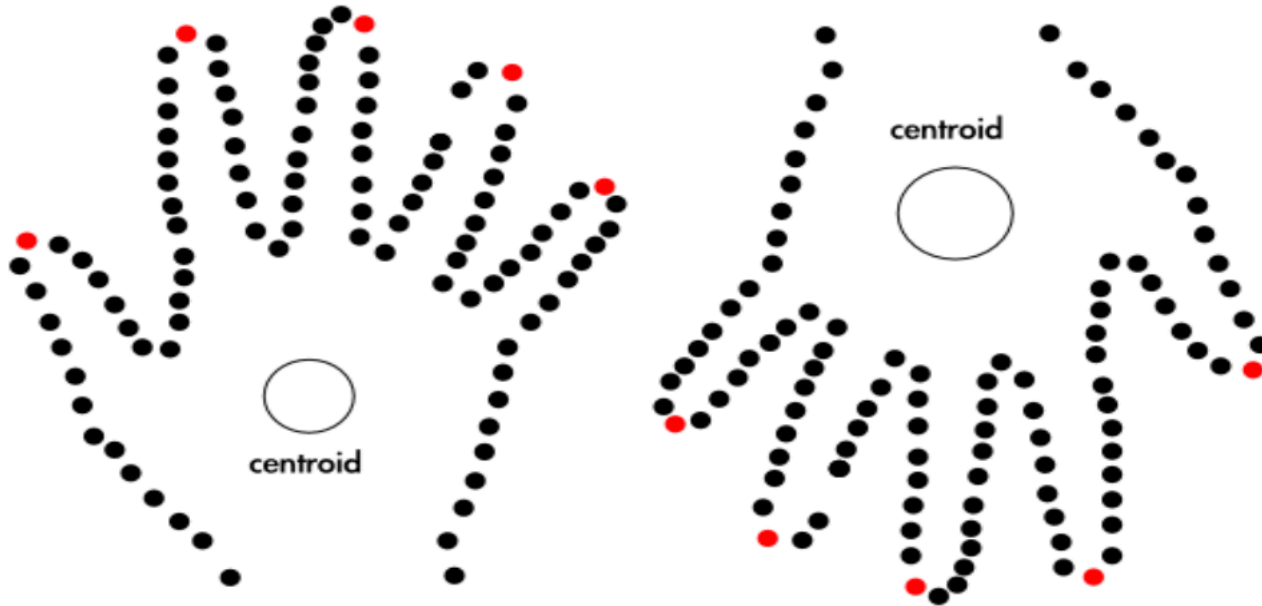
red = point in question

green = k away



# Approach 2: k-Curvvature

- To distinguish between finger tips and the spaces between fingers we can keep only those points that are further from the centroid than some threshold.





# Approach 2: k-Curvature

- To distinguish between fingertips and the spaces between fingers we can keep only those points that are further from the centroid than points  $k$  away.
- **Alternatively:** Compute the cross product between  $[C_i(j), C_i(j - k)]$  and  $[C_i(j), C_i(j + k)]$ . Depending on the sign of the cross product, we can determine if the point is a fingertip or a space between fingers.

# References

[1] Royal T., “Two Fingertip Tracking Techniques: k-curvature and convexity defects” . Tutorial. (August 2012)

<http://www.tmroyal.com/a-high-level-description-of-two-fingertip-tracking-techniques-k-curvature-and-convexity-defects.html>

[2] Neethu A. Real-time Static and Dynamic Hand Gesture Recognition. International Journal of Scientific & Engineering Research 7, 9 (March 2013)

<https://www.ijser.org/paper/Real-Time-Static-and-Dynamic-Hand-Gesture-Recognition.html>