METHODS

IMAGE ANALYSIS

Data acquisition and basic analysis. Videos were recorded by INSERT-MODEL-HERE using CAMERA-MODEL cameras. Videos extracted using SOFTWERE and analyzed by custom made MATLAB (Mathworks, R2013a) code which uses the image processing toolbox for extracting velocity, area and path. Recording from speakers input were recorded by a custom made system (Elaboration? Figure? Something??).

A visual representation of the speakers activity. Recorded information was integrated graphically into the videos by our code. Each speaker is represented by a circle in proximity to its original location as it appears in the video. An active speaker appears in a red color in a shade relative to the speaker intensity. The red shades are divided into 10 levels. Assumed maximal speaker intensity of 900V, thus every red level represents 90V. The higher the intensity, the stronger the red color.

Calculating aspect-ratio. We used the following algorithm to calculate the aspect ratio of the spot.

```
initialize max-ratio to 0
set degrees to 1

while degrees is less than or equal to 90, increased by 3

        rotate picture at current number of degrees
        create a cross in the center of the robot  %center was calculated earlier
        % the cross is always straight, while the image is rotating

        horizontal-line  <- rotated image ∩ horizontal-cross-line
        vertical-line    <- rotated image ∩ vertical-cross-line

        if      vertical-line is longer  than horizontal-line
                temp    <- vertical-line divided by horizontal-line
        else
                temp    <- horizontal-line divided by vertical-line
        end if

        if      temp  > max-ratio
                max-ratio       <- temp
        end if

        degrees         <- degrees + 3;

end loop

save max-ratio for current frame
```

Calculating center, area and velocity. As a preprocessing step before analysis we converted each frame to a binary image to distinguish between the spot and the background. After detecting the spot we were able to use MATALB image processing toolbox built-in functions to find its *center* and its *area* in each frame. Those functions get a binary image of objects and return the objects' centers in a 2D coordinate, and their area in pixels.

For calculating *velocity per second* we sampled two centers from two consecutive seconds and calculated the Euclidean distance between them. For calculating *area per second* and *center per second* we averaged the areas the centers, respectively, obtained for an entire second. This appears in the following pseudo-code:

```
Centers contains spot's center (2D coordinate) in each frame
Areas contains spot's area (in pixels) in each frame

Set FPS (frames per second) to the frame rate of the video
Set counter to 1

While counter is less than video's number of seconds

    % set frames indexes to the start and the end of the current second
    %i.e if FPS = 25 and counter = 2 then start index = 26 and end index = 50
    start-index     <- (counter-1)*FPS +1;
    end-index       <- (counter)*FPS;

    % calculate velocity (pixels/sec)
    current-coordinate      <- Centers[start-index]
    next-coordinate         <- Centers[end-index + 1]
    distance        <- Euclidean distance between current and next coordinates
    save distance for current second

    % calculate average centers
    average-center          <- average of Centers[from start-index to end-index]
    save average-center for current second

    % calculate average area
    average-area            <- average of Areas[from start-index to end-index]
    save average-area for current second

    counter         <- counter + 1;

End loop
```