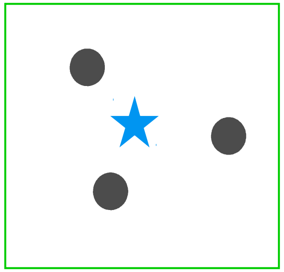
Probabilistic Sound Localization in Wireless Acoustic Sensor Networks

Alexander Wallar
Juan Ye

Dr. Juan Ye
School of Computer Science
University of St Andrews, UK
juan.ye@st-andrews.ac.uk

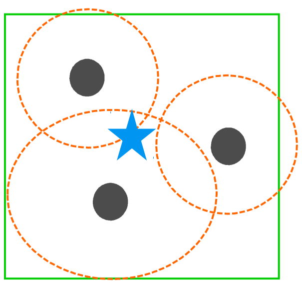
Process



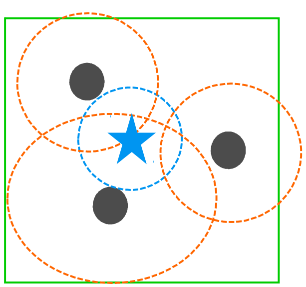
Acoustic nodes:

- (x,y)
- Timestamp
- Sound fingerprint
- Sound pressure level (spl)

Classification



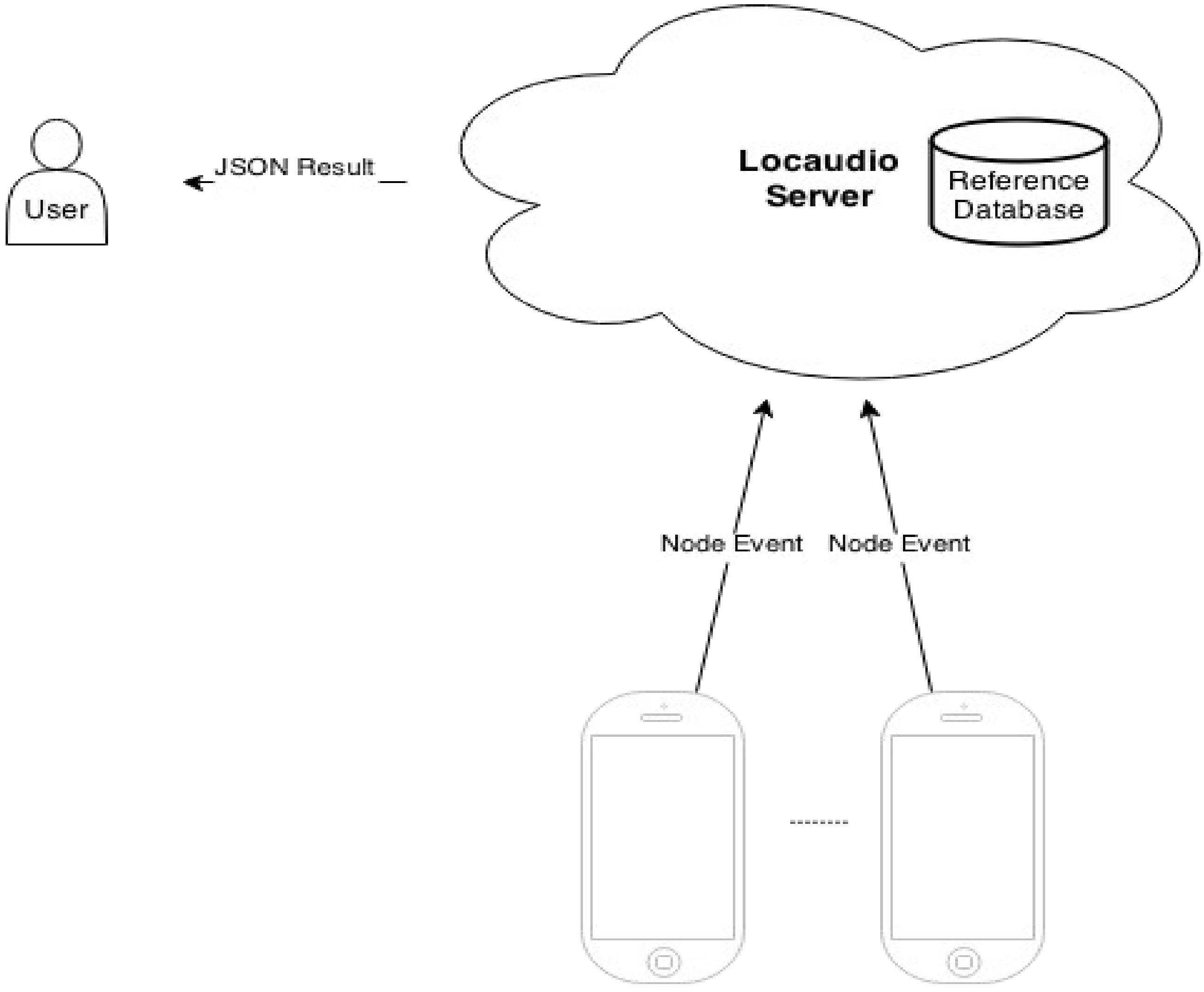
- Classify name of sound
- Retrieve reference spl and radius



Localisation

- Identify the area of sound source with
  - Radius
  - Probability

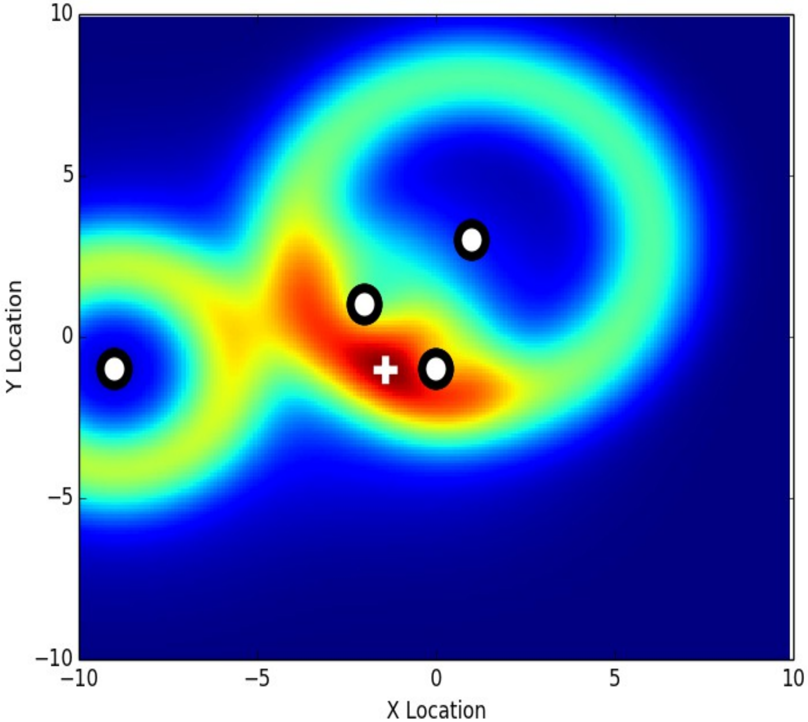
Process

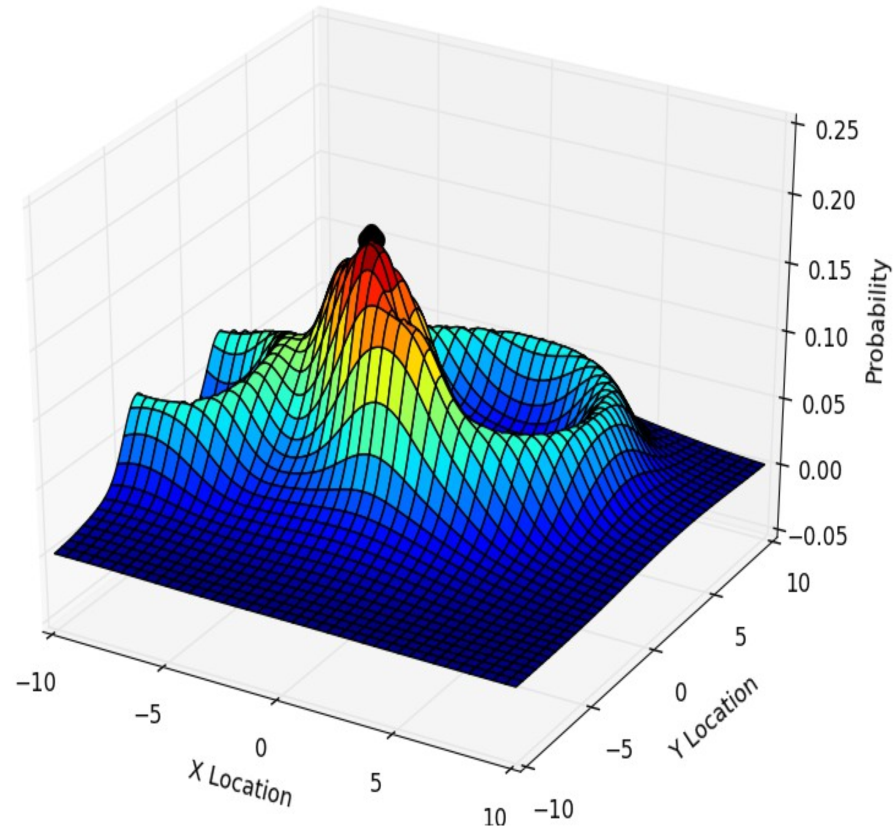

User
JSON Result
Locaudio Server
Reference Database
Node Event
Node Event
....

Sound classification

- We use audio fingerprinting
- Similar to what Shazam uses [Wang]
- Fingerprint determination from sound is done on the node (phone)
- Fingerprints are sent to the server for classification

Localization





Localization

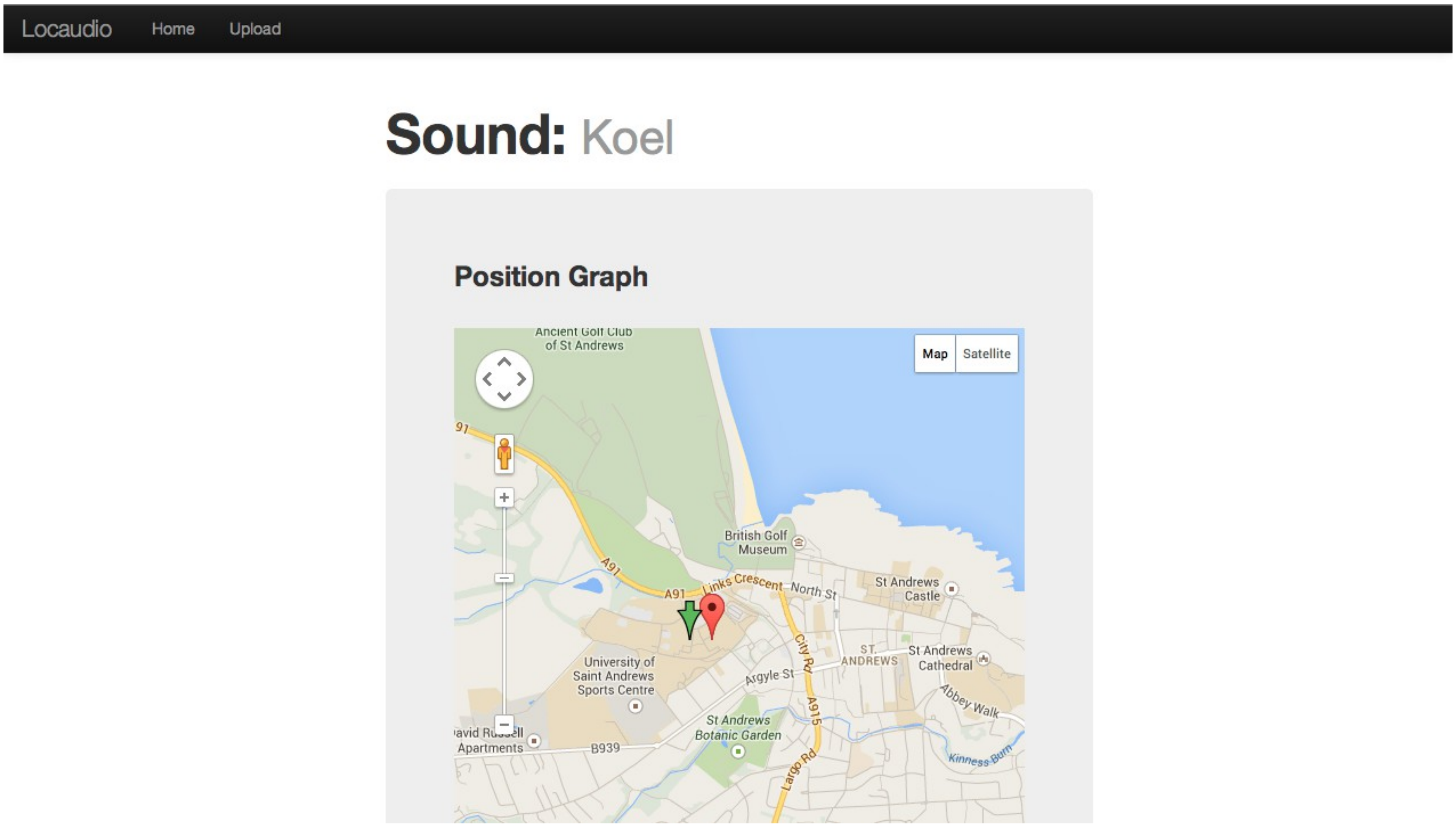
$$\mathrm{D}_s(r, spl, spl') = r \cdot 10^{\frac{(spl - spl')}{20}}$$

$$\mathrm{D}(x, y, n) = \sqrt{(x - n.x)^2 + (y - n.y)^2}$$

$$\mathcal{P}(x, y, r, spl, events) = \frac{\sum_{e \in events} \mathcal{N}(\mathrm{D}(x, y, e), \mathrm{D}_s(r, spl, e.spl), \textsc{GetSD}(e, events))}{|events|}$$

$$\mathcal{L}(r, spl, events) = \underset{x,y}{\arg\min} \; -\mathcal{P}(x, y, r, spl, events)$$

Interactive display



Potential applications

- Detecting and tracking animals

- Optimize sensor deployment
  - Where to deploy sensors to achieve
    - Minimal number of nodes
    - Cover sufficient areas
    - Maximize localization accuracies

Thank you!

Q&A

aw204@st-andrews.ac.uk
http://st-andrews.academia.edu/AlexWallar

Technical details :: Multiple Source

- Bias the gradient decent of the probability function
- Use affinity propagation to cluster local minimas

$$\mathcal{L}_m(r, spl, events, x_g, y_g) = \underset{x,y}{\textsc{GradientDecent}} \left(-\mathcal{P}, x_g, y_g, r, spl, events\right)$$

Algorithm 2 Multiple Source Sound Localization

1: $\textsc{GetLocations}(r, spl, eventList) \rightarrow$
2: $locations \leftarrow \textsc{InitLocationList}()$
3: **for all** $e \in eventList$ **do**
4: $\quad locations.add(\mathcal{L}_m(r, spl, eventList, e.x, e.y))$
5: $centers \leftarrow \textsc{AffinityPropagation}(locations)$ $\{centers$ is a list of (x, y, c) points where c is the confidence$\}$
6: **return** $centers$

Technical details :: Loudness vs. Distance

- Equation seen before: $\mathrm{D}_s(r, spl, spl') = r \cdot 10^{\frac{(spl - spl')}{20}}$
- Distance from a sound is related to the sound pressure level
- Given reference data, we can determine the distance from the sound