# Sensing and topology: some ideas by other people, and an early experiment

Simon Dobson, Danilo Pianini, and Mirko Viroli

University of St Andrews
Scotland UK

simon.dobson@st-andrews.ac.uk

Università di Bologna
Cesena IT

danilo.pianini@unibo.it
mirko.viroli@unibo.it

University of St Andrews | FOUNDED 1413

# Introduction

There are some fundamental problems with sensing

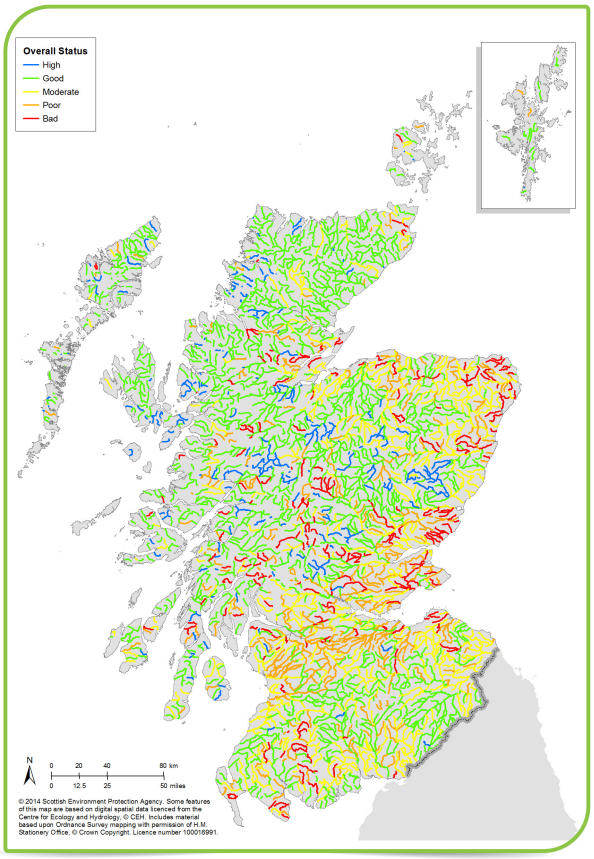
- How good are the readings we get?
- Are they good enough to support the conclusions we want to draw from them?
- How should our confidence change as the sensors age?
- Can we work in a fully distributed environment?

This talk

- Some thoughts on the foundations of sensing
- A mathematical approach (by others)
- An experiment in applying this maths (by us)

# Overview

# Example: water sensing – what we think we've got




Source: Scottish Government, 2012

- Perfect, clean data
- …at arbitrary resolution
- …completely up-to-date

# Example: water sensing – what we've actually got


Source: MapRoom

- Noisy spot data
- …taken at discrete points
- …at various times in the past

# An idealisation – 1

Move to an ideal situation

- Find a mathematical description for the "best" situation we could possibly find ourselves in for sensing
- …even if we can't build it – ignore engineering realities

Constraints

- More sensors is (usually) better
- Better resolutions, more frequent observations, …
- …so we want **lots** of **great** sensors

# An idealisation – 2

An infinitely dense field of infinitely small sensors

- We can now see arbitrarily small phenomena
- We actually always want summaries (at some resolution) of this idealisation

# An idealisation – 2

An infinitely dense field of infinitely small sensors

- We can now see arbitrarily small phenomena
- We actually always want summaries (at some resolution) of this idealisation

This is the **continuum limit**

- It can't get any better…
- A scenario in which we can do integration

# Making it precise – 1

## Sensor field

- One sensor at each point of $X \subset \mathbb{R}^2$
- Each sensor can sense the water quality (or whatever) $Q_x$ for every $x \in X$

## Sensing

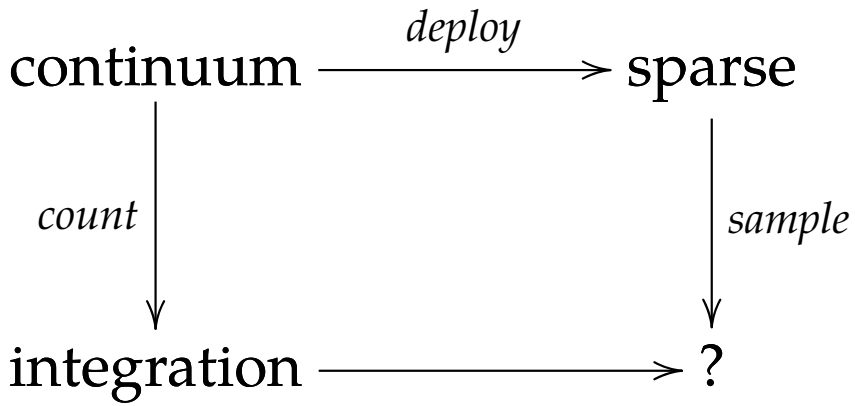
- We're looking for an average over an arbitrarily small region
- Suppose we look at a region of radius $r$ around a point $x$, which we denote as $S$
- Then the average quality $Q_S$ is given by:

$$Q_S = \frac{1}{\pi r^2} \int_S Q_x dx$$

# Back to reality – 1

So far so imaginary…

- A foundation for what we **want** to do
- In moving to what we **can** so, we reduce the number and density of the sensors
- Want to migrate the maths alongside



# Back to reality – 2

Sparse sensors

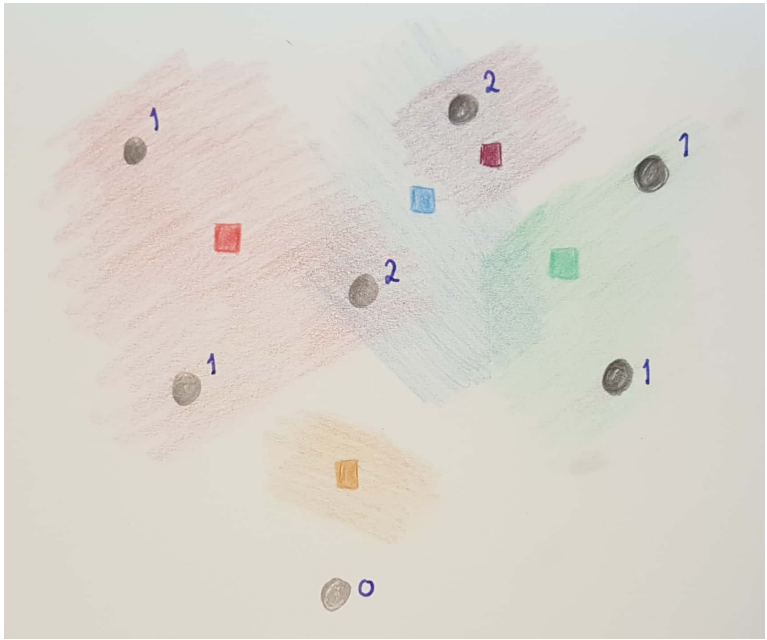
- A finite number, positioned sparsely
- The exact geometry may be uncertain

A mathematical framework to inject our previous calculations into

- An approach other than familiar integration
- The rest of this talk is about how we might do this

# Overview

# The target counting problem




How many targets are there? [1]

- Sensors *count* but can't *identify* targets
- Several *different* sensors may be observing the *same* target

[1] D. Wu, B. Zhang, H. Li, and X. Cheng. Target counting in wireless sensor networks. In H. Ammari, editor, **The art of wireless sensor networks, volume 2: Advanced topics and applications**. Springer-Verlag, 2014. ISBN 978-3-642-40065-0

# Target counting in an ideal world – 1

Our infinitely dense sensor field $X$ counts targets

- Each target $a \in A$ has an impact on a subset $U_a \subset X$ of the sensors

Define a "height" at each sensor in terms of an indicator function

$$\mathbb{I}_{U_a}(x) = \begin{cases} 1 & x \in U_a \\ 0 & \text{otherwise} \end{cases} \Rightarrow h(x) = \sum_a \mathbb{I}_{U_a}(x)$$

- If each $U_a$ is a circle of radius $r$ then each target is seen by $\pi r^2$ sensors

# Target counting in an ideal world – 2

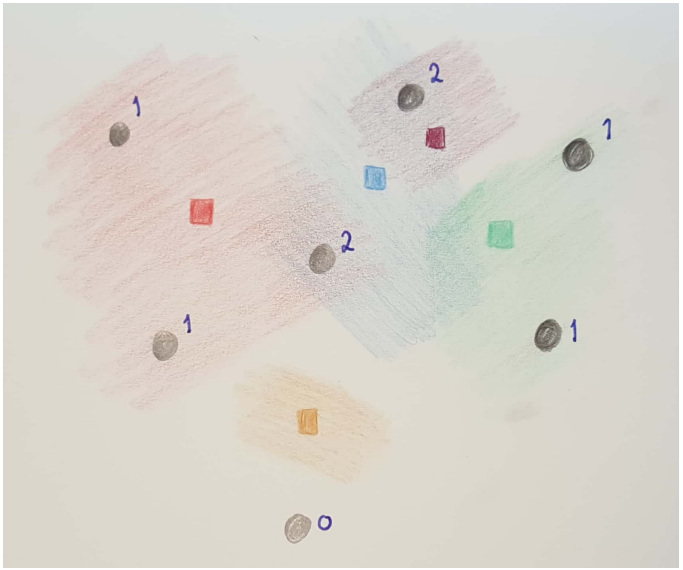
Then the number of targets is given by $\#A = \frac{1}{\pi r^2} \int_X h(x)\,dx$ since [2]

$$\begin{aligned}\int_{\mathbb{R}^2} h(x)\,dx &= \int_X \sum_a \mathbb{I}_{U_a}(x)\,dx \\ &= \sum_a \int_X \mathbb{I}_{U_a}(x)\,dx \\ &= \sum_a \pi r^2 \\ &= \pi r^2 \#A\end{aligned}$$

[2] Y. Baryshnikov and R. Ghrist. Target enumeration via Euler characteristic integrals. **SIAM Journal of Applied Mathematics**, 70(3):825–844, 2009. URL http://dx.doi.org/10.1137/070687293

# Target counting in reality

- A discrete set of counts
- Not independent observations, but correlated with each other in some way
- "Close together" sensors may be seeing the same target



# Dealing with discrete counts

Can we capture the structure of the dataset collected?

- Link together sensors that might co-observe a target
- Use this consistency structure to try to even-out ambiguities

This approach goes under the name of **topological data analysis** [3]

- Use the tools of algebraic topology to capture the "shape" of the dataset
- Analyse it to look for features like holes in the sampling
- Perform operations like integration

[3] F. Chazal and B. Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. Technical Report arXiv::1710.04019, arXiv, October 2017. URL https://arxiv.org/abs/1710.04019

# Simplicial topology – 1

## Topology

- Algebraic analysis
- Smoothness
- Properties stable under large perturbations

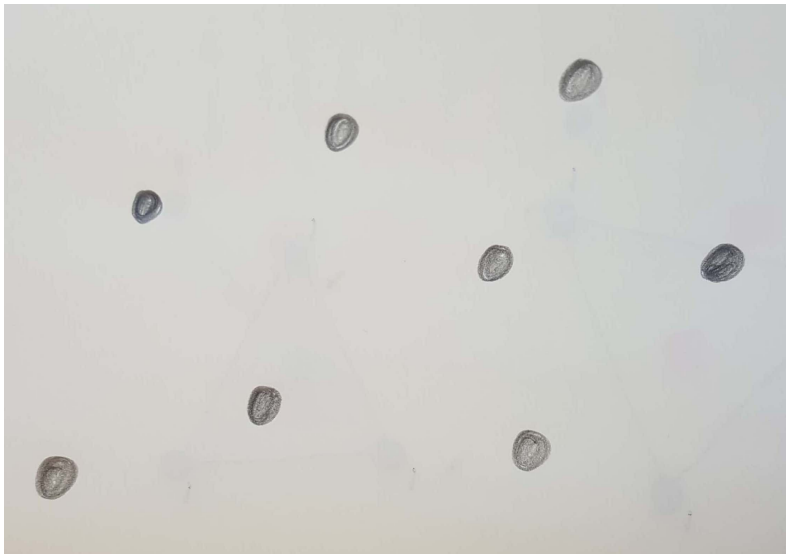
> *Topology! The stratosphere of human thought! In the twenty-fourth century it might possibly be of use to someone…*
>
> Alexander Solzhenitsyn, The First Circle

## *Simplicial* topology

- A generalisation of the idea of a graph [4]
- Points, line, triangles, tetrahedra, …

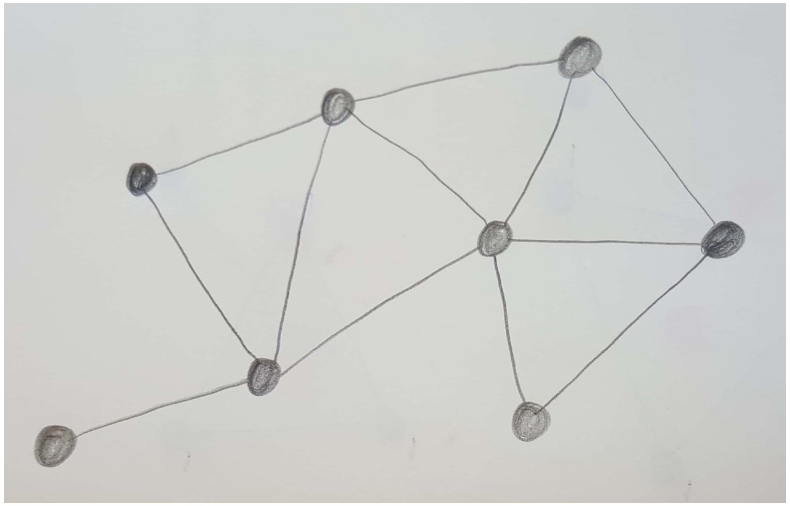
[4] H. Edelsbrunner. **A short course in computational geometry and topology**. SpringerBriefs in Applied Sciences and Technology. Springer, 2014. ISBN 978-3-319-05956-3. URL doi://10.1007/978-3-319-05957-0

# Simplicial topology – 2



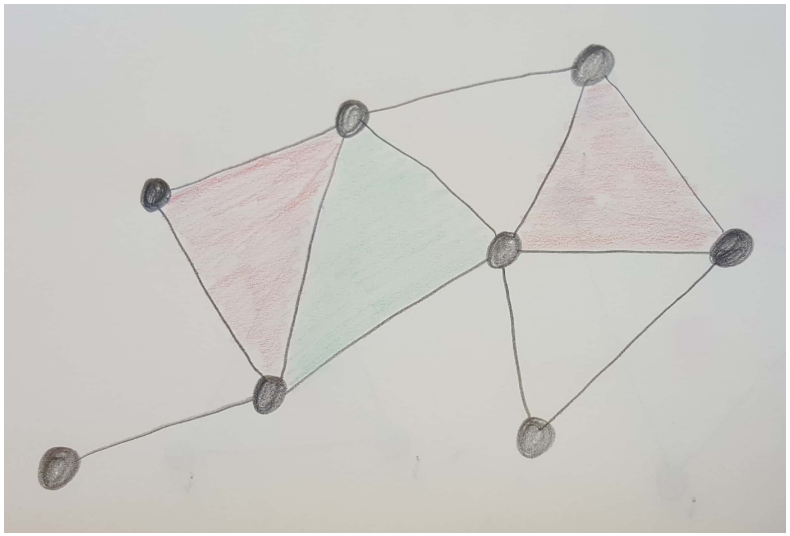
- A collection of points
- Zero-dimensional
- Each is an **0-simplex**

# Simplicial topology – 2



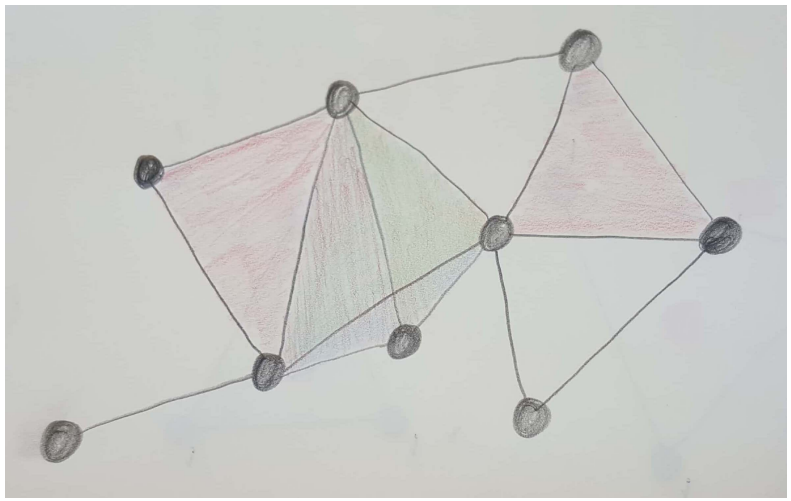
- 1-simplices

# Simplicial topology – 2



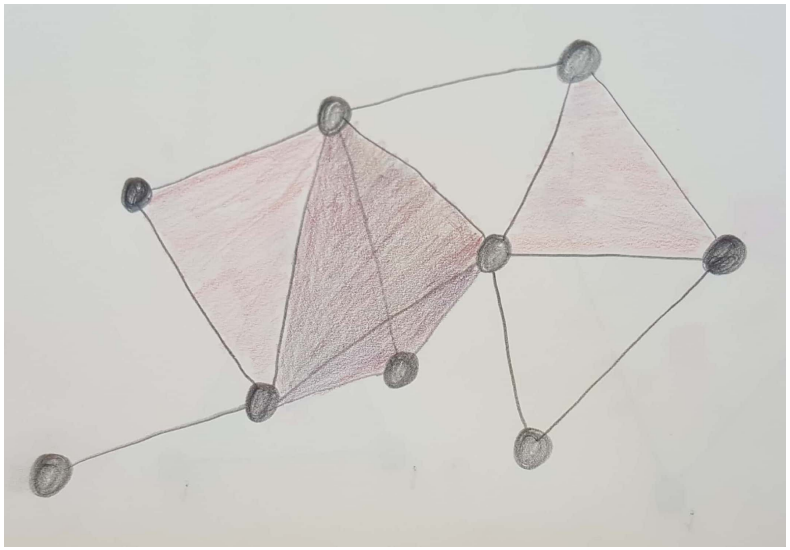
- 2-simplices

# Simplicial topology – 2



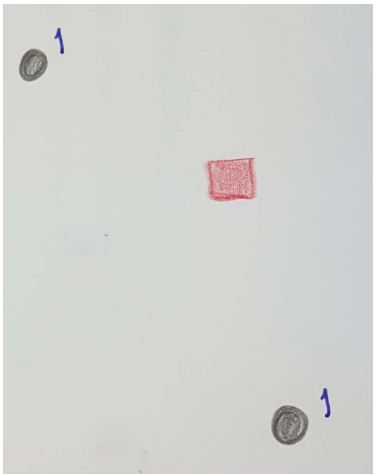
- Surrounding a void…

# Simplicial topology – 2



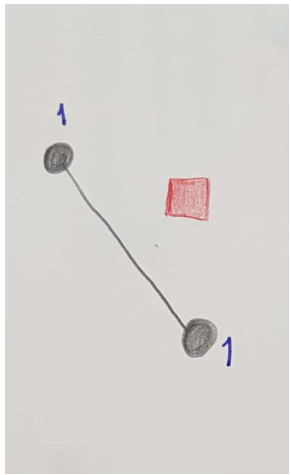
- …that we then fill with a 3-simplex
- …and so on for higher simplices

# Target counting – consistency – 1




Two distant sensors

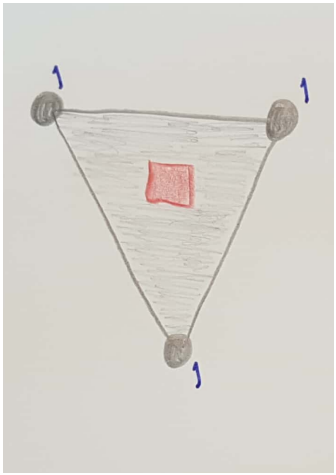
- Must be different targets




Two distant sensors

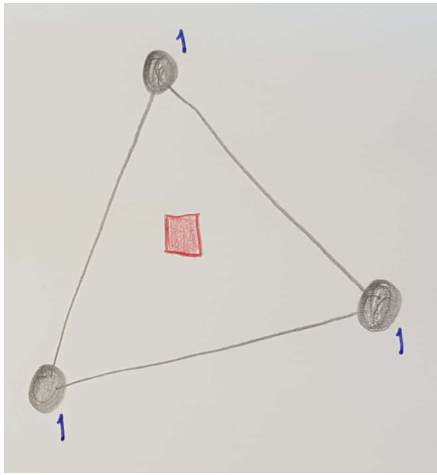
- Could be the same target
- Consistency between observations

# Target counting – consistency – 2




Three sensors with consistent observations

- Strongly suggests that we're looking at the same target




Three sensors with pairwise consistency

- A "hole" in the coverage
- May be looking at different targets

# Overview

# Pause for context

All the mathematical ideas that have gone before already existed in the literature

- Our interest was in using them to build distributed sensor systems
- **Self-stabilising** algorithms using local computations
- A new style of algorithms, well-founded in topology

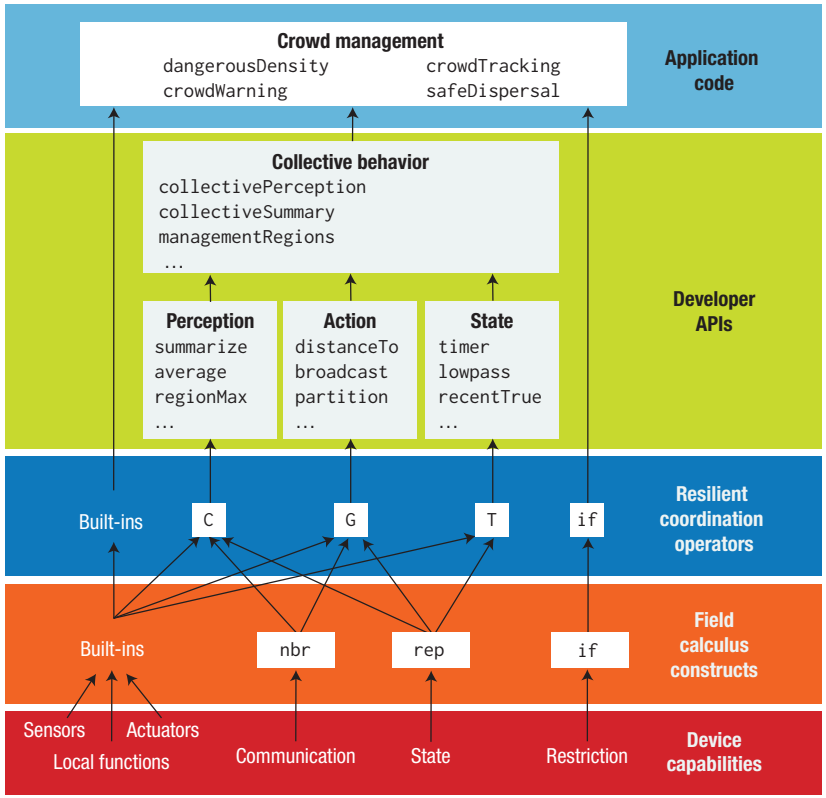
So we went and built it

- Field calculus, a framework for self-stabilising systems
- Look for new operators and patterns to support this new style of algorithm

# Field calculus

A formal basis for self-stabilising systems [5]

- Distributed data structure of computational fields, map devices to values
- "Building block" operators that compose smoothly
- Portable across a range of devices




[5] J. Beal, M. Viroli, D. Pianini, and F. Damiani. Self-adaptation to device distribution changes. In G. Cabri, G. Picard, and N. Suri, editors, **10th IEEE International Conference on Self-Adaptive and Self-Organizing Systems, SASO 2016**, pages 60–69, September 2016. doi: 10.1109/SASO.2016.12

# Target counting – local counts

Each sensor $p$ (an 0-simplex of $S$) maintains a count $h(p)$ of targets it can see

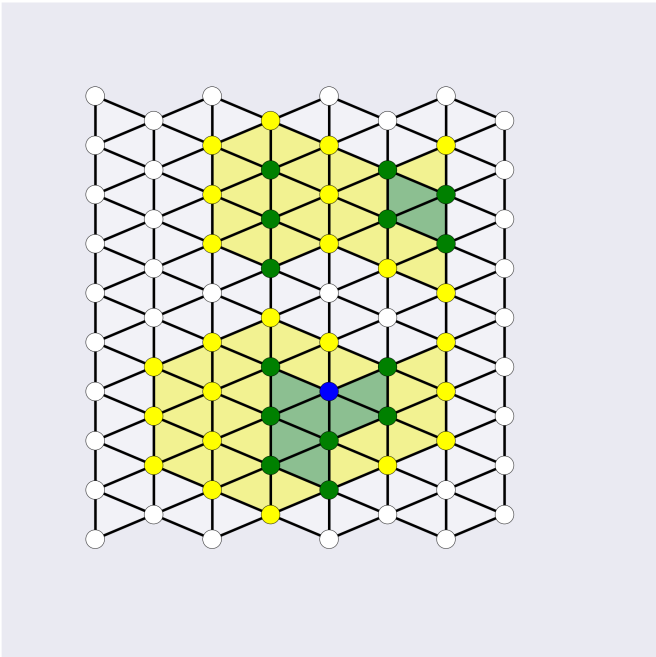
Introduce a set $A$ of targets

- Each target $a \in A$ has an impact $U_a$ on the sensor field
- Define $h(p) = \#\{a | p \in U_a\}$
- Extend $h$ to all simplices by $h(p) = min(\{h(q) | q \in \mathbb{B}(p)\})$

The target counting problem

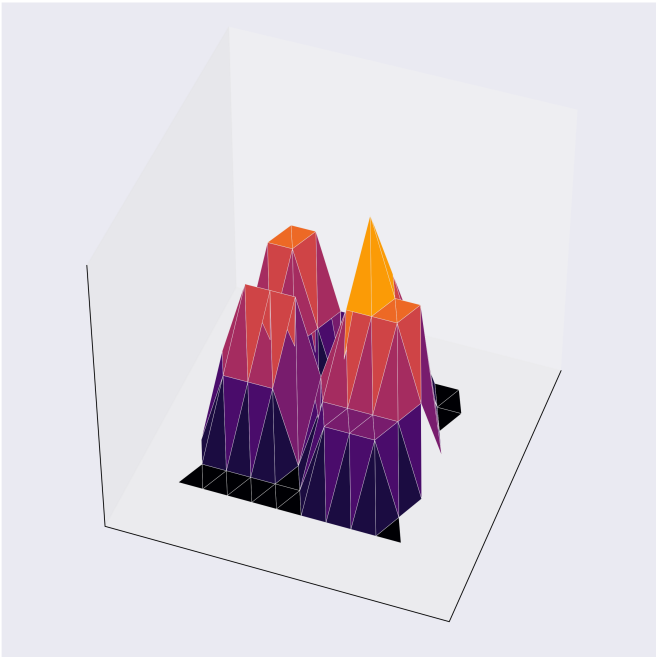
- Form an estimate of $\#A$ using only the values of $h(p)$ at each 0-simplex $p$

# Target counting – "flooding the landscape"



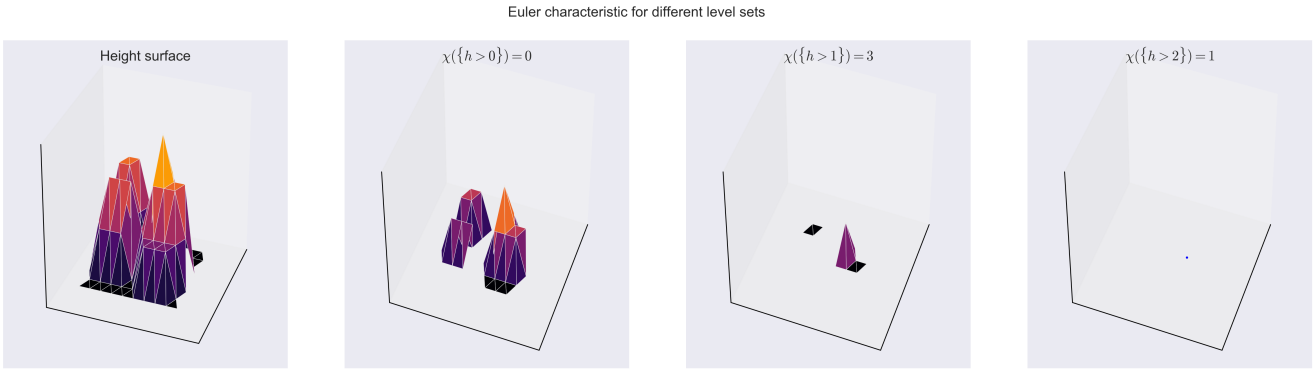
- Counts at each sensor (0-simplex)

# Target counting – "flooding the landscape"



- Interpret counts as heights
- Extended to the higher simplices
- Form a "landscape"

# Target counting – "flooding the landscape"




Flood the landscape

- Define **level sets** $\{h > s\}$ containing all simplices $p \in S$ such that $h(p) > s$
- Construct a sequence of level sets for $0 \leq s < \infty$

# The Euler characteristic of a simplicial complex

$$\chi(S) = \#vertices - \#edges + \#triangles - \#tetrahedra + \ldots$$

Functions as a kind of hole detector

- A triangulated plane has $\chi = 1$
- A plane with $n$ holes has $\chi = 1 - n$
- Two isolated triangulated "islands" have $\chi = 2$
- …as do three "islands", one of which has a hole…

# Target counting – Euler integration

We can now estimate the target count [6] as

$$\begin{aligned} \#A &= \int_S h \, d\chi \\ &= \sum_{s=0}^{\infty} \chi(\{h > s\}) \end{aligned}$$

- Simplicial complex encodes enough information to reject (some) duplicate observations
- Estimate becomes exact in the continuum limit

[6] Y. Baryshnikov and R. Ghrist. Target enumeration via Euler characteristic integrals. **SIAM Journal of Applied Mathematics**, 70(3):825–844, 2009. URL http://dx.doi.org/10.1137/070687293

# Design sketch

Define the estimator as a function over a computational field [7]

- Fully distributed
- Continuous approximation of the target count without additional mathematical machinery

Re-define the Euler characteristic in terms of a distributed **Euler contribution** $\chi^p$

- Computable locally from $\#S_k^p$, the number of $k$-simplices of which sensor $p$ is part
- Sum over $k$ (locally) and $p$ (using a field)

[7] D. Pianini, S. Dobson, and M. Viroli. Self-stabilising target counting in wireless sensor networks using Euler integration. In **Proceedings of the Eleventh IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO'17)**, pages 11–20, September 2017. URL http://dx.doi.org/10.1109/SASO.2017.10

# Shape of the design – 1

Compute $\chi^p$ locally at each sensor node

- A single local calculation at each node

Slice the network into level sets to sum the values

- Create "views" of the network that directly capture the level sets $\{h > s\}$, and sum over them

# Shape of the design – 2

Four phases

1. Aggregate computation of local network structure
2. Aggregate "slicing" of network into level sets
3. Local computation of local Euler contribution
4. Aggregation of result to an elected leader and re-broadcast to all nodes

Implemented using Protelis [8]

- Direct transcription of the design into field calculus

[8] D. Pianini, M. Viroli, and J. Beal. Protelis: practical aggregate programming. In **Proceedings of the 30th Annual ACM Symposium on Applied Computing**, pages 1846–1853, 2015. doi: 10.1145/2695664.2695913

# Implementation – local computation

```
def static double chi( Map<DeviceUID,Tuple> n ){
  Sets.powerSet(n.keySet).groupBy[it.size]
  .mapValues[it.filter[set |
   set.forall[device |
    n.getOrDefault(device, createTuple)
     .containsAll(set.reject[it==device])
   ]
  ]
  .mapValues[it.size as double]
  .map[(-2 * (it.key % 2) + 1) * it.value / (it.key + 1)]
  .reduce[$0 + $1]
}
```

# Implementation – aggregation

## Build the level sets

```
def slices(height) {
  map(range(1, height), h -> { chi(neighborsNeighbors()) })
}

def localContribute(targets) {
  slices(targets).reduce(0, sum)
}
```

## Run the program

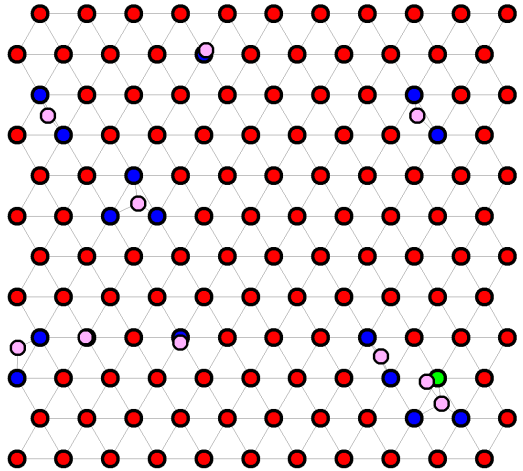
```
merge(localContribute(localTargetCount()))
```

# Overview

# Simulations

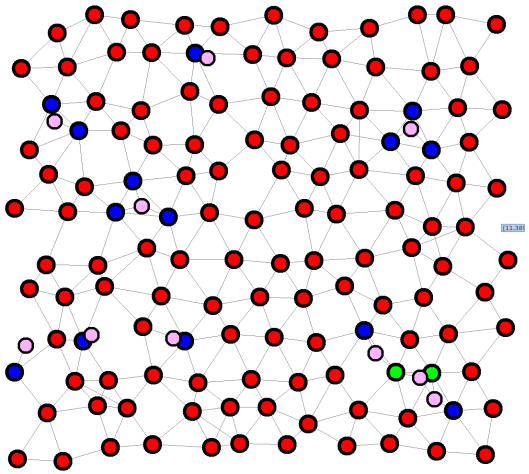
Simulated [9] an area $10x10m$ with different sensor deployments

- Square and hexagonal lattices, regular and perturbed
- Vary device density, targets to track, detection range
- **Sensing/communications ratio** (SCR)



Compute average normalised RMS error of estimate at equilibrium

$$E_{RMS} = \frac{\sum_{d=0}^{N} \left(\frac{T-c_d}{T}\right)^2}{N}$$

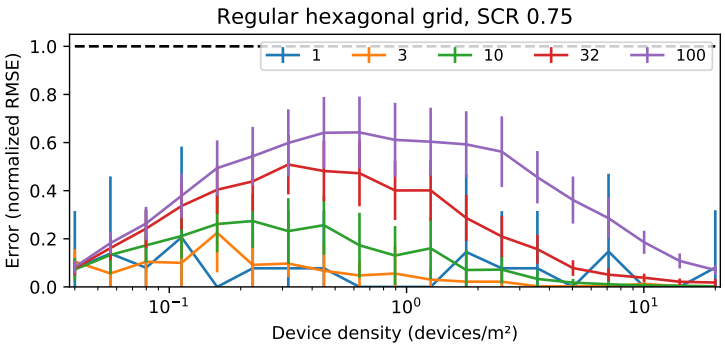


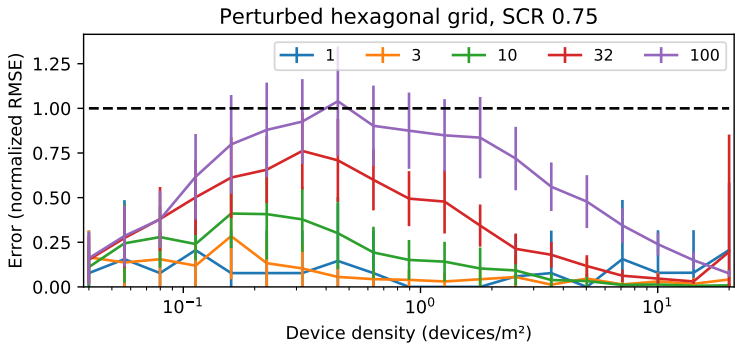
[9] D. Pianini, S. Montagna, and M. Viroli. Chemical-oriented simulation of computational systems with ALCHEMIST. **J. Simulation**, 7(3):202–215, 2013. doi: 10.1057/jos.2012.27

# Results – device density

Raising the sensor density isn't always immediately beneficial

- Counting initially gets *worse* for *higher* densities
- Easy to count tightly-packed *or* widely-dispersed targets

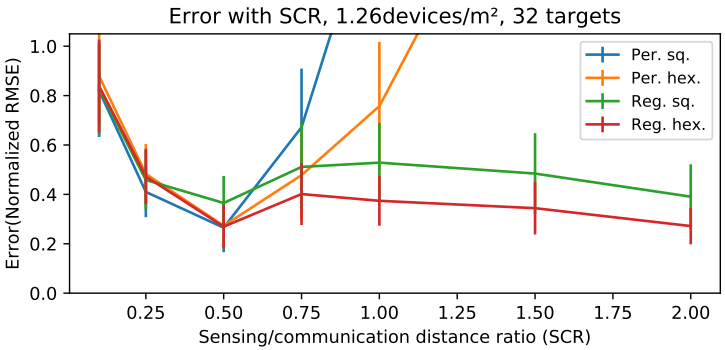


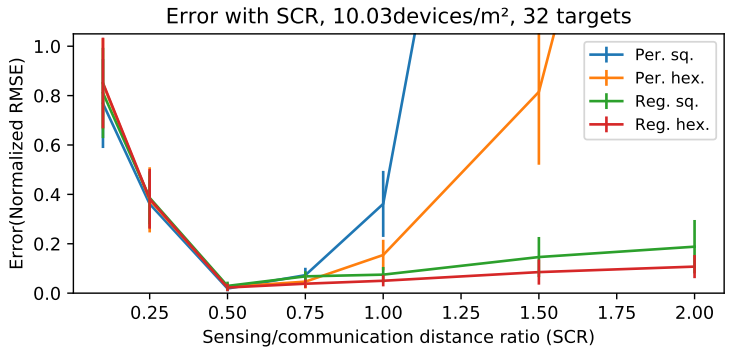



# Results – SCR

## SCR defines coverage

- Small SCR introduces lots of holes; large SCR increases duplicate counting
- Sweet spot at about $SCR = [0.5, 0.75]$ (as you might expect)



# Observations

Estimate quality dependent on the interaction between sensor field and targets

- We can explain **local** over- and under-counting
- Sensitive to regularity of the network
- Tuning of SCR

But we don't understand the **global** implications

- What constitutes a "good enough" triangulation?
- How do higher simplices (more correlated observations) affect the estimate?

# Overview

# What we learned

The topological approach formalises the transition to sparse sensors

- A well-founded way to do sensor fusion
- Generalises to a larger body of mathematics, **sheaf theory**

Embed and analyse the real sensing problems we encounter

- Transient (noise) or systematic (drifting) mis-counting?
- What happens as the sensors fail, and the simplicial complex becomes sparse and hole-y?
- What exactly is happening with the interactions we've observed?

# Thank you

> *We must bear in mind, then, that there is nothing more difficult or dangerous or more doubtful of success than an attempt to introduce a new order of things.*
>
> Niccoló Machiavelli, The Prince (1515)

## Additional material

- Code and simulation `https://bitbucket.org/danysk/experiment-2017-saso-counting`
- Python library for computing with simplicial topology `https://pypi.python.org/pypi/simplicial`

# References


Y. Baryshnikov and R. Ghrist. Target enumeration via Euler characteristic integrals. **SIAM Journal of Applied Mathematics**, 70(3):825–844, 2009. URL `http://dx.doi.org/10.1137/070687293`.

J. Beal, M. Viroli, D. Pianini, and F. Damiani. Self-adaptation to device distribution changes. In G. Cabri, G. Picard, and N. Suri, editors, **10th IEEE International Conference on Self-Adaptive and Self-Organizing Systems, SASO 2016**, pages 60–69, September 2016. doi: 10.1109/SASO.2016.12.

F. Chazal and B. Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. Technical Report arXiv::1710.04019, arXiv, October 2017. URL `https://arxiv.org/abs/1710.04019`.

H. Edelsbrunner. **A short course in computational geometry and topology**. SpringerBriefs in Applied Sciences and Technology. Springer, 2014. ISBN 978-3-319-05956-3. URL `doi://10.1007/978-3-319-05957-0`.

D. Pianini, S. Montagna, and M. Viroli. Chemical-oriented simulation of computational systems with ALCHEMIST. **J. Simulation**, 7(3):202–215, 2013. doi: 10.1057/jos.2012.27.

D. Pianini, M. Viroli, and J. Beal. Protelis: practical aggregate programming. In **Proceedings of the 30th Annual ACM Symposium on Applied Computing**, pages 1846–1853, 2015. doi: 10.1145/2695664.2695913.

D. Pianini, S. Dobson, and M. Viroli. Self-stabilising target counting in wireless sensor networks using Euler integration. In **Proceedings of the Eleventh IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO'17)**, pages 11–20, September 2017. URL `http://dx.doi.org/10.1109/SASO.2017.10`.

D. Wu, B. Zhang, H. Li, and X. Cheng. Target counting in wireless sensor networks. In H. Ammari, editor, **The art of wireless sensor networks, volume 2: Advanced topics and applications**. Springer-Verlag, 2014. ISBN 978-3-642-40065-0.

# Addendum: How can counting go wrong? – 1

Re-visit the Euler integral

$$\#A = \sum_{s=0}^{\infty} \chi(\{h > s\})$$

Suppose we add a new target. There are two ways that $\#A$ can observe this new target:

1. Increase maximum value of $h$ by 1
2. Split a level set into two, which will then increase $\chi(\{h > s\})$ by 1

# Addendum: How can counting go wrong? – 2

It's possible for *neither* or *both* of these to happen

- Under- or over-counting
- Depends on the interaction between the target and the sensors that observe it