Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)

# From Raw Sensor Data to Detailed Spatial Knowledge

**Peng Zhang** and **Jae Hee Lee** and **Jochen Renz**
Australian National University
Canberra, Australia
{p.zhang, jae-hee.lee, jochen.renz}@anu.edu.au

## Abstract

Qualitative spatial reasoning deals with relational spatial knowledge and with how this knowledge can be processed efficiently. Identifying suitable representations for spatial knowledge and checking whether the given knowledge is consistent has been the main research focus in the past two decades. However, where the spatial information comes from, what kind of information can be obtained and how it can be obtained has been largely ignored. This paper is an attempt to start filling this gap. We present a method for extracting detailed spatial information from sensor measurements of regions. We analyse how different sparse sensor measurements can be integrated and what spatial information can be extracted from sensor measurements. Different from previous approaches to qualitative spatial reasoning, our method allows us to obtain detailed information about the internal structure of regions. The result has practical implications, for example, in disaster management scenarios, which include identifying the safe zones in bushfire and flood regions.

## 1 Introduction

There is the need for an intelligent system that integrates different information such as sensor measurements and other observations with expert knowledge and uses artificial intelligence to make the correct inferences or to give the correct warnings and recommendations. In order to make such an intelligent system work, we need to monitor areas of interest, integrate all the available knowledge about these areas, and infer their current state and how they will evolve. Depending on the application, such an area of interest (we call it a spatial region or just region) could be a region of heavy rainfall, a flooded region, a region of extreme winds or extreme temperature, a region with an active bushfire, a contaminated region, or other regions of importance.

Considering such a region as a big blob or as a collection of pixels on a screen is not adequate, because these regions usually have a complex internal structure. A bushfire region for example, might consist of several areas with an active fire, several areas where the fire has passed already but are still hot, several areas that are safe, or several areas where there has been no fire yet, etc. These areas form the components of the bushfire region. Knowing such detailed internal structure of a region can help predict how the region evolves, as well as for determining countermeasures to actively influence how a region evolves.

In this paper, we present methods for

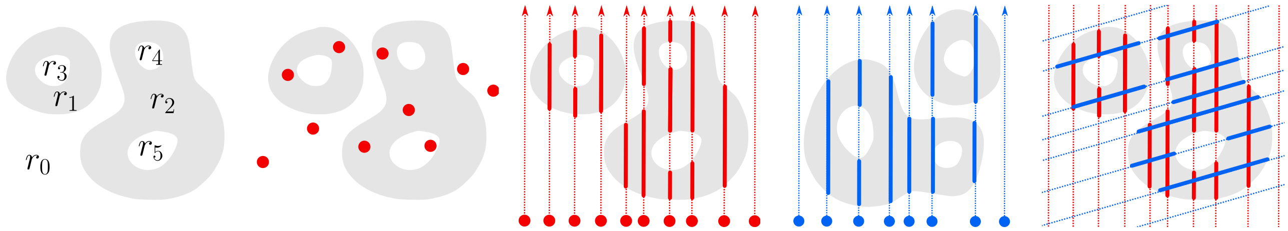
- integrating measurements of spatial regions from different sensor networks to obtain more accurate information about the regions;
- extracting spatial information from the integrated sensor measurements in the form of a symbolic representation.

Having a symbolic representation of the internal structure instead of a collection of pixels on a screen allows us to answer queries, derive new knowledge, identify patterns, or to express conditions and rules, for example, when a warning must be issued. It also allows us to obtain an accurate approximation in the first place: by being able to represent the internal structure of a spatial region, we can build up a partial representation of the internal structure of a monitored region, track the region over time, and refine the partial representation whenever we obtain new sensor measurements or other new information.

The remainder of this paper is organised as follows: in Section 2 we motivate and formally define the problem in question. This is followed by a review of related literature in Section 3. In Section 4 we describe the methods for matching sensor measurements and for extracting spatial information and evaluate them in Section 5. We conclude the paper in Section 6.

## 2 Our Approach

The problem we are trying to solve, i.e. extracting accurate spatial information about the *internal structure* of a region using sensor measurements, is a very general problem with many facets. It depends on (1) what assumptions we make about the regions, (2) what assumptions we make about the sensors, and (3) what kind of spatial information we are considering. We will make some restrictions on all three points to see if the desired outcome (i.e. accurate spatial information) can be achieved at all. In future work we plan to lift these restrictions and to allow greater generality. The restrictions and assumptions we make in this paper are the following:




(a) A complex region consisting of positive components $r_1, r_2$ and negative components $r_0, r_3, r_4, r_5$.

(b) A complex region moving over a set of arbitrarily placed sensors on the plane.

(c) Normalised sensor measurement of a complex region. Thick lines represent positive intervals and thin lines negative intervals.

(d) Normalised sensor measurement of the same region using a different set of sensors.

(e) Integrating two measurements reveals partial information about which positive intervals belong to the same components.

Figure 1: A complex region measured by two sets of sensors.

**Regions:** A *complex region* $R$ is a two-dimensional spatial entity in the plane that can have a finite number of *components*. Each component can have arbitrary shape, extent and location and can have a finite number of *holes* of arbitrary shape, extent and location. Each hole can recursively contain a complex region which is part of the original complex region. We say that each component, independent of whether it is inside a hole or not, is a *positive component* of $R$, while each hole and each bounded or unbounded area of the plane that is not a positive component is a *negative component* of $R$. See Figure 1a for an example of such a region.

*Restrictions on regions:* We assume that a given complex region is rigid, i.e. does not change shape, extent or relative location of any of its components. We allow regions to move, to change speed, direction of movement and their orientation. We further assume that during a sensor measurement, regions have constant speed, direction and orientation.

**Sensors:** A *sensor* $s$ is a point in the plane that can measure whether it is inside a region or outside a region at time $t$, that is it senses 1 if $s$ is in a positive component of $R$ and 0 if $s$ is in a negative component of $R$. As a practical example, consider a sensor that measures (absence of) direct sunlight. If a positive component of a region is between the sun and the sensor it measures 1, otherwise 0. We consider sets $S$ of sensors $s_i$ located on arbitrary but known locations on a plane (see Figure 1b).

*Restrictions on sensors:* We assume that sensors are static and do not change their locations. Sensor plane and region plane can be different, but must be parallel.

**Sensor measurements:** A single sensor measurement (in short: a *measurement*) $M$ occurs when a complex region $R$ moves over a set $\{s_1, \ldots, s_n\}$ of sensors. Given the above assumptions and restrictions on regions and sensors, each measurement $M$ corresponds to a set $\{\ell_1, \ldots, \ell_n\}$ of parallel directed lines, where each line $\ell_i$ corresponds to one sensor $s_i$. The lines (more precisely line segments) start at the beginning of the measurement $M$ and end at the end of the measurement. Each line may contain zero or more alternating *positive* and *negative intervals*, where each positive interval corresponds to the time when the sensor was in a positive component of $R$ and each negative interval corresponds to when the sensor was outside of $R$ (see Figure 1c). We assume that each measurement covers all positive components of the region. Note that the ordering of the parallel lines and the distances between lines depend on the direction of movement of the measured region and the actual location of the sensors on the plane relative to this direction.

A *normalised measurement* $M$ is a measurement where all sensors are located on a line orthogonal to the direction of movement of the measured region, with arbitrary distance between sensors. Any measurement can be converted to its corresponding normalised measurement $M$ if the direction of movement of the measured region and the actual sensor locations are known. This can be achieved by the following steps: (1) select a line orthogonal to the direction of movement such that all sensors are on the side of the line where the region is coming from, (2) project all sensors orthogonally on this line, and (3) subtract the distance between the line and each sensor $s_i$ at the beginning of each $\ell_i$.

*Restrictions on sensor measurements:* We assume that the direction of movement of regions is known, which can be easily obtained if there is at least one sensor consisting of multiple pixels. Then we can always obtain a normalised measurement. Note that this is independent of the orientation of a region, i.e. how it is rotated, which is unknown. We further assume that different measurements of the same region are independent, in particular that distances between sensors of a normalised measurement vary. This could result from different sets of sensors or from a different direction of movement.

**Consequences of our assumptions and restrictions:** Our assumptions are relatively general, the strongest restriction we make is that complex regions are rigid. Except for direction of movement and location of sensors, everything else is unknown and unrestricted. Given two independent normalised measurements, these measurements correspond to two sets of parallel lines that cut through the measured region at independent and unknown angles $\delta$ (due to unknown orientation of regions). Depending on the speed during a measurement, regions appear stretched/skewed along the direction of movement. The slower the movement, the more stretched is the appearance of the region as each sensor stays within the region longer. In addition, due to the unknown distance between sensor plane and region plane, the relative scale of each measurement is also unknown. In the example of a direct sunlight sensor, regions closer to the sensors appear larger in the measurement. Note that under our assumptions, it is interchangeable to have moving regions and static sensors, or moving sensors and static regions.

**Spatial information:** The spatial information we are considering is the topology of $R$ (for a complex region this refers to the number of components and holes and their hierarchy; cf. [Li, 2010]), the relative size of the components of $R$, the relative direction and the relative distance between components of $R$.

**Problem description and motivation:** In the following, we take as input two different measurements $M$ and $M'$ as sets of parallel lines obtained in the way described above. The distance between the parallel lines is arbitrary and it does not matter which sensor corresponds to which line. That means that the only input we have available are the parallel lines and their intervals, we do not know what is happening between these lines. In order to obtain accurate spatial information about $R$ we need to be able to infer this missing information. This will be achieved by integrating the different measurements through reconstructing the actual relationship between the measurements and the region.

Our approach is motivated by the following observations:

1. Given a single measurement $M$ where two adjacent parallel lines have overlapping positive intervals $i_1$ and $i_2$, i.e. there is a time when both sensors are in a positive component of $R$. Without knowledge about the area between the two adjacent lines, we cannot say for sure whether $i_1$ and $i_2$ refer to the same positive component of $R$ or to two different ones (see Figure 1c).
2. Given a second measurement $M'$ that measured $R$ using a different angle $\delta'$ (see Figure 1d). Assume there is a positive interval $i_3$ in $M'$ that intersects both $i_1$ and $i_2$, then we know for sure that $i_1$ and $i_2$ and also $i_3$ are part of the same positive component of $R$ (see Figure 1e).

It is clear that by integrating two or more measurements we obtain much more accurate information about a region than having one measurement alone. However, obtaining this information is not straightforward as the angle and the scale of the measurement is unknown. This results from our assumption that distance between region and sensors and orientation of the measured region are unknown. Therefore, our first task is to develop a method to identify a translation, rotation, and scaling that allows us to convert one measurement into another such that the intervals can be matched. Once we have obtained a good match, we will then extract detailed spatial information from the integrated sensor information. We then evaluate our method and determine how close the resulting spatial information is to the ground truth and how the quality depends on factors such as sensor density.

## 3 Related Work

Although the setting of our problem seems related to classical sensor fusion techniques [Crowley, 1993; LaValle, 2010], no such techniques are known to be adequate to tackle our problem. For example, a sensor fusion method such as the extended Kalman filter requires a state transition model with specific parameters for each individual region and measurements of system inputs for its implementation, which is in practice difficult to obtain. Our approach is generic and does not have any of these requirements. In particular, it does not require a description of region geometry in terms of equations.

In the context of qualitative spatial reasoning [Cohn and Renz, 2008], most research has taken the objects (e.g. regions or relations) of a symbolic reasoning process as *given* and paid only little attention to the problem of acquiring such information from the environment. In [Santos and Shanahan, 2003] a stereo vision sensor measures 2D depth profile of a 3D environment taken at a particular height. The objects in the environment are represented as peaks of the depth profile graph. The depth peaks from different snapshots are then matched to provide a qualitative interpretation of their transitions. Inglada and Michel [2009] use remote sensing to extract topological relations between complex objects. A graphical representation of the relations is built to allow recognising objects using graph-matching techniques. Both mentioned methods, however, are not suitable for capturing detailed internal structures of regions that result from environmental parameters, such as light, sound, chemicals, temperature, pressure or magnetism. Our work makes no assumptions about the sensor type and includes sensor networks based on microelectromechanical system (MEMS) [Gardner *et al.*, 2005] sensors that overcome such deficiencies.

Similar to our work there are approaches that employ a MEMS based sensor network: Worboys and Duckham [2006; 2013] describe a computational model for tracking topological changes in spatial regions monitored by a sensor network. Jiang and Worboys [2009] go further and provide a systematic formal classification of the possible changes to a region, based on point-set topology and graph homomorphism. All these algorithms, however, assume the regions to be constructed from connected subgraphs of adjacent sensor nodes that detect the region, which do not allow us to reason about the internal structure of the areas between the sensors measurements. By contrast, this paper considers all aspects of the internal structure of regions and uses it to infer knowledge about the areas between sensor measurements.

Finally, there are interpolation techniques from computer vision [Szeliski, 2010] and spatial statistics [Cressie, 2011], which are not suitable for situations where the sensor network is sparse and only partial information about regions without further statistical information is known.

## 4 Closing the Gap in Sensor Measurements

In this section we present a method for integrating two independent sensor measurements of a complex region (Section 4.1) which allows us to extract more precise spatial information about the complex region (Section 4.2).

### 4.1 Integration of Sensor Measurements

For integrating two independent measurements we search for parameter values for translation $\tau$, rotation $\rho$ and scaling $\sigma$

(a) An overlay of two sensor measurements with mismatches that are indicated by circles.

(b) A partition of the measurement from Figure 1c, which consist of two clusters $c_1$ and $c_2$.

(c) A partition of the measurement from Figure 1d, which consist of two clusters $c'_1$ and $c'_2$.

(d) Initial match based on the spatial structures of the clusters. There are 4 mismatches.

(e) Result after fine-tuning with a local search.

Figure 2: Integration of sensor measurements based on Algorithm 1.

of the second measurement that *minimise* mismatches with the first measurement. A mismatch occurs if, in the overlay of the sensor measurements, a positive interval from one measurement intersects with a negative interval from another measurement (see Figure 2a). An uninformed local search in the parameter space will likely lead to local minima. In order to avoid such local minima, we first build clusters in the individual measurements that give us information about spatial structures of the measurements (Figure 2b and Figure 2c), and exploit this information to obtain a good initial match (Figure 2d). Once good initial values for $\tau, \rho, \sigma$ are found, we fine-tune and improve the match by using a local search method that further reduces the number of mismatches between the measurements (Figure 2e).

The details of our procedure are given in Algorithm 1. The main function is IntegrateSensorMeasurements, which takes two sensor measurements and first determines for each sensor measurement a set of possible partitions (lines 2–3) by calling GenPartitions. A partition $P$ of a measurement $M$ consists of clusters, where a cluster is again comprised of positive intervals in $M$ that potentially belong to the same positive component of underlying region $R$ (see Figure 2b and Figure 2c). To be precise, we say two positive intervals $i_1$ and $i_2$ are neighboured and write $i_1 \sim i_2$, if $i_1$ and $i_2$ are from adjacent sensor lines and temporally overlap. We furthermore say that positive intervals $i_1$ and $i_2$ belong to the same cluster (or to the same equivalence class), if $i_1 \sim^+ i_2$, where $\sim^+$ is the transitive closure of $\sim$. This allows us to obtain a partition (or quotient set) $M/{\sim^+}$ of the sensor measurement $M$.

Since some of these clusters could actually belong to the same component of $R$ (unless there exists a negative interval in $M$ that clearly separates the clusters), we consider all coarser partitions of $M/{\sim^+}$ induced by combining clusters that could potentially belong to the same connected component of $R$. This is done by calling Function GenCoarserPartitions (line 16), where two clusters are merged if the distance between the clusters is less than a threshold value. This value controls the number of possible coarser partitions which is at most $2^{|M/{\sim^+}|}$. Note that it is irrelevant if the detected clusters reflect the positive components of $R$ truthfully, as they are only used to find a good initial match and not for obtaining spatial information.

From collections $L, L'$ of all coarser partitions of measurements $M, M'$ that we obtained in the previous step, we now try to find a pair of partitions from $L \times L'$ that will lead to the best match of $M, M'$. As a heuristic for finding such a pair we rank all pairs $(P, P') \in L \times L'$ according to the structural similarity between $P$ and $P'$ by means of Function StructDist (line 4). More precisely, Function StructDist measures the structural distance between partitions $P$ and $P'$, for which we consider the relative sizes and relative directions (i.e. angles) of the clusters in each partition. Given two lists of clusters $(c_1, c_2, \ldots, c_k)$ of $P$ and $(c'_1, c'_2, \ldots, c'_{k'})$ of $P'$ sorted according to their sizes (the lower the index, the bigger the size), we determine angular information of salient clusters (the lower the index, the more salient is the cluster), and compare the similarity between $P$ and $P'$. This is achieved by the following formula:

$$\sum_{i=1}^{\max\{k,k'\}-2} |\angle c_i c_{i+1} c_{i+2} - \angle c'_i c'_{i+1} c'_{i+2}| \cdot w_i, \qquad (1)$$

where $w_i$ is a weight given by the maximum of the sizes of $c_i$ and $c'_i$. The formula in (1) captures the angles between salient clusters and returns a large value, if the the angles between salient clusters of the two partitions are dissimilar, and returns a smaller value if the angles between salient clusters are similar.

After ranking the pairs of partitions according to the similarity criteria in formula (1), we start with the pair with the highest similarity from the ranked list $S$, and calculate initial parameter values $\tau, \rho, \sigma$ respectively for translation, rotation and scaling. To overcome local minima we choose the pair of partitions with the next highest similarity from $S$ until either the number $v$ of mismatches are below a threshold $\epsilon$ or no more pairs of partitions are left in $S$ (lines 6–10). Obtaining the number $v$ of mismatches and the parameter values from a pair of partitions is achieved by calling function GetParameter (line 8), which compares the two most salient and clearly distinguishable clusters $(c_1, c_2)$ and $(c'_1, c'_2)$ from the partitions and determines the translation, rotation and scale between those pairs. After obtaining the initial parameters, function MinMismatches further reduces the number of mismatches and fine-tunes the result by performing a local search around the initial parameter values (line 11). This is done by means of point set registration, which is a local search technique for matching 2D point sets [Jian and Vemuri, 2011]. As

**Algorithm 1:** Integrating sensor measurements.

**Function** IntegrateSensorMeasurements(*M, M'*)
  **Input** : Two different measurements $M$ and $M'$.
  **Output**: The merged version of $M$ and $M'$.
  `// initial match`
  $L \leftarrow \mathsf{GenPartitions}(M)$
  $L' \leftarrow \mathsf{GenPartitions}(M')$
  $S \leftarrow \mathsf{Rank}(L \times L', \mathsf{key} = \mathsf{StructDist})$
  $v^* \leftarrow \infty, \quad \tau^* \leftarrow 0, \quad \rho^* \leftarrow 0, \quad \sigma^* \leftarrow 0$
  **while** $v^* > \epsilon$ *and* $S \neq \emptyset$ **do**
    $(P, P') \leftarrow \mathsf{Pop}(S)$
    $(v, \tau, \rho, \sigma) \leftarrow \mathsf{GetParameter}(P, P')$
    **if** $v < v^*$ **then**
      $(v^*, \tau^*, \rho^*, \sigma^*) \leftarrow (v, \tau, \rho, \sigma)$
  `// local search`
  $(\tau^*, \rho^*, \sigma^*) \leftarrow \mathsf{MinMismatches}(M, M', \tau^*, \rho^*, \sigma^*)$
  **return** $M \cup \mathsf{Rotate}_{\rho^*}(\mathsf{Translate}_{\tau^*}(\mathsf{Scale}_{\sigma^*}(M')))$

**Function** $\mathsf{GenPartitions}(M)$
  **Input** : A sensor measurements $M$.
  **Output**: A list $L$ of partitions of $M$.
  **foreach** *positive intervals* $i_1, i_2 \in M$ **do**
    Set $i_1 \sim i_2$, if $i_1$ and $i_2$ are from adjacent sensors and temporally overlap.
  $L \leftarrow \mathsf{GenCoarserPartitions}(M/{\sim^+})$
  **return** $L$

**Function** $\mathsf{StructDist}(P, P')$
  **Input** : Two sets $P, P'$ consisting of clusters.
  **Output**: The structural distance between $P$ and $P'$.
  $(c_1, c_2, \ldots, c_k) \leftarrow \mathsf{SortSize}(P)$
  $(c'_1, c'_2, \ldots, c'_{k'}) \leftarrow \mathsf{SortSize}(P')$
  **return** $\sum_{i=1}^{\max\{k,k'\}-2} |\angle c_i c_{i+1} c_{i+2} - \angle c'_i c'_{i+1} c'_{i+2}| \cdot w_i$

**Algorithm 2:** Extracting spatial knowlege.

1 **Function** $\mathsf{ExtractSpatialKnowledge}(M)$
  **Input** : Sensor measurement $M$.
  **Output**: An augmented containment tree of $M$.
2   $C \leftarrow \mathsf{GetConnectedComponents}(M)$
3   $T \leftarrow \mathsf{ContainmentTree}(C)$
4   $T^+ \leftarrow \mathsf{Augment}(T)$
5   **return** $T^+$

6 **Function** $\mathsf{GetConnectedComponents}(M)$
  **Input** : Sensor measurement $M$
  **Output**: List of connected pos. and neg. components
7   **foreach** *positive intervals* $i_1, i_2 \in M$ **do**
8     Set $i_1 \sim i_2$, if $i_1$ intersects $i_2$.
9   **foreach** *negative intervals* $i_1, i_2 \in M$ **do**
10     Set $i_1 \sim i_2$, if $i_1$ intersects $i_2$.
11   **return** $M/{\sim^+}$

point set registration requires two sets of points as its input, we uniformly sample points from positive sensor intervals of both of the sensor measurements. The final outcome is the union of two sensor measurements, where the second sensor measurement is transformed using the parameters found.

## 4.2 Spatial Knowledge Extraction

After integrating sensor measurements, it is now possible to extract more precise spatial information. For example, from only one sensor measurement it was impossible to extract precise information about the connected components of a complex region, as all positive intervals are disconnected. From the merged sensor measurements, however, it is possible to extract this information, as overlaying different measurements connects the intervals that were previously disconnected. In what follows we present an algorithm (Algorithm 2) for extracting spatial information from the merged sensor measurements.

The main function ExtractSpatialKnowledge takes as its input a merged sensor measurement and extracts spatial information. The extracted spatial information is represented in the form of a tree, which is called an *augmented containment tree*, where each node is either a positive component or a negative component and the parent-child relation is given by the containment relation of the components, i.e. a negative (resp. positive) component is a child of a positive component (resp. negative component), if the former is inside the latter. Additionally, for each node we provide spatial relations (directions, distances, and sizes) between its children nodes and between parent and children nodes. Most other relations can be inferred or they can be extracted as well. This allows us to obtain information about the internal structure of any connected component of a complex region. Which spatial relations we extract is actually not relevant anymore as we now have a reasonably fine grained actual representation of the complex region which allows us to obtain spatial relations of most existing spatial representations.

Function ExtractSpatialKnowledge first determines connected components in the sensor measurement (line 2). We not only detect positive components, but also negative components (i.e. holes), as holes play a role in several applications as mentioned in the introduction, e.g. disaster scenarios. We first identify the class of all intervals that are connected by the intersection relation $\sim$, and obtain the partition $M/{\sim^+}$, where $\sim^+$ is the transitive closure of $\sim$ (line 11).

From the connected components we build a tree that reflects the containment relations between the components (line 3). We first detect the root $r_0$ of the tree, i.e. the negative component in the background which contains all other components. This can be easily done by finding the first negative interval of any sensor measurement and returning the negative component that contains the negative interval. Afterwards, we recursively find the children nodes of a parent node $r_i$, which are components "touching" $r_i$. This too can be realised easily by determining the interval (and the component containing the interval) that is immediately next to an interval of the parent component. We generate the contain-

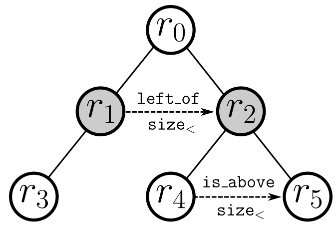



Figure 3: An illustration of the augmented containment tree extracted from the merged measurements in Figure 2e.

Figure 4: A randomly generated complex region.

ment tree using breadth-first search.

Once the containment tree is built, we augment the tree with spatial relations from the spatial calculi that are most suitable for a given application. We recursively detect the internal structure of each component by going through every node of the containment tree using breadth-first search and extract spatial information (directions, distances, and sizes) of its children. The final outcome is then an augmented containment tree (see Figure 3). Having such a tree, one can answer spatial queries such as "$\texttt{right\_of}(x, A) \wedge \texttt{size}_<(x, B)$" by finding nodes in the tree that satisfy the queries. These relations can then be used for spatial reasoning in the standard way [Cohn and Renz, 2008].

## 5 Evaluation

The purpose of this paper was to obtain accurate information about the internal structure of a complex region by integrating sensor measurements. In order to evaluate the quality and accuracy of our method, we developed a tool that can randomly generate arbitrarily complex spatial regions (see Figure 4). The generated regions are initially bounded by polygons, but the edges are then smoothened. We also randomly generate a line of sensors (corresponding to a normalised measurement), with random orientation and gap between sensors and random speed and take measurements for each choice. The regions are $800 \times 600$ pixels with 8–9 components on average.

For our experiments, we selected equidistant sensors, since we want to analyse how the gap between sensors influences the accuracy of the outcome. It does not affect our method. The chosen orientation, scale and speed of the sensor measurement is unknown to our method and will only be used to compare accuracy of the resulting match with the ground truth. We are mainly interested in how good the resulting match is to the correct match. The closer orientation and scale is to the true values, the more accurate the spatial information we can extract, provided the sensors are dense enough. A perfect match (i.e. orientation and scale differences are 0) leads to the best possible information for a given sensor gap. We therefore measure the difference in orientation and scale to the correct values. We also measure the percentage of mismatches of all possible intersections. While orientation and scale differences can only be obtained by comparing to the ground truth, mismatches can be obtained directly. The denser the sensors, the more mismatches are possible.

We created 100 random complex regions and two measurements with random angle per region and per sensor density and applied our method to match the measurements and to extract the spatial relations from the resulting match. We then evaluated the effectiveness of the initial match and the additional fine-tuning step by treating them separately, and also compared the result against the point set registration method from [Jian and Vemuri, 2011], which is used for the local search in our algorithm.

As shown in Table 1, the initial matches before the fine-tuning with local search are already close to the correct match and are significantly better than point set registration alone. However, the number of mismatches is as expected fairly large for sparser sensor density (i.e. higher gap size). This is significantly reduced by adding the fine-tuning step and could be further reduced by using more advanced local search heuristics. The point set registration method requires more computational resources when the sensor network is dense, because there are more points to match. By contrast, the runtime in the initial match increases with growing gap size. This is because sparse sensors lose more information than dense sensors. The possibility for building different combinations of clusters of a measurement increases as the gap size grows, which significantly increases the runtime.

The number of components detected by our algorithm is generally greater than the ground truth. There are two main reasons for this observation: first, some components are only connected by very slim parts which are usually missed by the sensors; second, some mismatches causes some very small components to be detected. However, the difference between the number of components we detected and the ground truth is reasonable and can be improved by integrating further measurements.

## 6 Conclusions

In this paper we presented a sophisticated method for obtaining accurate spatial information about the internal structure of complex regions from two independent relatively sparse sensor measurements. The task was to develop a method that can infer information about what happens in the areas the sensors cannot sense by integrating different independent measurements of the same complex region. The intention was to obtain more information than just the sum of individual measurements, each of which does not contain much accurate information. Furthermore, we wanted to obtain a symbolic representation of the sensed region that allows intelligent processing of the knowledge we obtained, such as query processing, change detection and prediction, or the ability to specifying rules, for example when warnings are issued. One of the benefits of our method is that it is very generic and does

Table 1: Evaluation based on 100 randomly generated complex regions.

| Sensor Gap (px) | PSR [a] | | Initial Match | | | | Initial Match + Local Search | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | MM [b] | RT | MM | AD | SD | RT | MM | AD | SD | RT |
| 5 | 48.95 | 1077685 | 1.83 | 0.03 | 0.73 | 46663 | 1.46 | 0.03 | 0.61 | 1126474 |
| 15 | 48.06 | 152179 | 8.68 | 0.12 | 1.49 | 91730 | 4.09 | 0.05 | 0.66 | 219884 |
| 20 | 49.27 | 83273 | 18.68 | 0.14 | 2.58 | 98244 | 6.19 | 0.05 | 0.99 | 147899 |
| 25 | 46.35 | 60832 | 22.36 | 0.16 | 2.69 | 133326 | 7.85 | 0.07 | 1.34 | 173744 |

[a] PSR: Point Set Registration in [Jian and Vemuri, 2011]
[b] MM: Mismatch (%), AD: Angle Difference (radian), SD: Scale Difference (%), RT: Runtime (ms)

not require any complicated state transition models or object parameterisations that are typically used in the area of sensor fusion.

As our evaluation demonstrates, our method can successfully integrate different independent sensor measurements and can obtain accurate spatial information about the sensed complex regions. We make some restrictions about the sensor setup and the complex regions we sense. In the future we plan to lift these restrictions and analyse if we can obtain similar accuracy when using a less restricted sensor setup, such as, for example, using mobile phones as sensors which can follow any path. Allowing non-rigid regions and to detect their changes also seems a practically relevant generalisation of our method.

In this paper we restricted ourselves to the detection of the internal structure of a single complex region. Obviously, this restriction is not necessary and we can easily apply our method to the detection of multiple complex regions and the relationships between the regions. In that case we can modify our sensors so they detect in which regions they are in at which time. Using the same procedure as for an individual complex region, we can obtain an integration of sensor measurements that allows us to infer the relations between multiple regions. What is not as straightforward is to change what sensors can measure, for example, a degree of containment, or continuous values rather than 1/0 values. Integrating such measurements requires significant modifications to our algorithms.

## Acknowledgments

We gratefully acknowledge helpful advice from Jochen Trumpf and Jason Jingshi Li. This research was supported by Australian Research Council (DP120103758).

## References

[Cohn and Renz, 2008] Anthony G. Cohn and Jochen Renz. Qualitative Spatial Representation and Reasoning. In *Foundations of Artificial Intelligence*, volume 3 of *Handbook of Knowledge Representation*, chapter 13, pages 551–596. Elsevier, 2008.

[Cressie, 2011] Noel A. C. Cressie. *Statistics for spatio-temporal data*. Wiley series in probability and statistics. Wiley, Hoboken, N.J, 2011.

[Crowley, 1993] James L. Crowley. Principles and Techniques for Sensor Data Fusion. In *Multisensor Fusion for Computer Vision*, number 99 in NATO ASI Series, pages 15–36. Springer Berlin Heidelberg, 1993.

[Duckham, 2013] Matt Duckham. *Decentralized Spatial Computing*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[Gardner *et al.*, 2005] Julian W. Gardner, Vijay K. Varadan, and Osama O. Awadelkarim. *Microsensors, MEMS, and smart devices*. Wiley, 2005.

[Inglada and Michel, 2009] Jordi Inglada and Julien Michel. Qualitative Spatial Reasoning for High-Resolution Remote Sensing Image Analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 47(2):599–612, 2009.

[Jian and Vemuri, 2011] Bing Jian and Baba C. Vemuri. Robust Point Set Registration Using Gaussian Mixture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1633–1645, August 2011.

[Jiang and Worboys, 2009] Jixiang Jiang and Michael Worboys. Eventbased topology for dynamic planar areal objects. *International Journal of Geographical Information Science*, 23(1):33–60, January 2009.

[LaValle, 2010] Steven M. LaValle. Sensing and Filtering: A Fresh Perspective Based on Preimages and Information Spaces. *Foundations and Trends in Robotics*, 1(4):253–372, 2010.

[Li, 2010] Sanjiang Li. A Layered Graph Representation for Complex Regions. In *Twelfth International Conference on the Principles of Knowledge Representation and Reasoning*, pages 581–583, April 2010.

[Santos and Shanahan, 2003] Paulo Santos and Murray Shanahan. A Logic-based Algorithm for Image Sequence Interpretation and Anchoring. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 1408–1410, San Francisco, CA, USA, 2003.

[Szeliski, 2010] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[Worboys and Duckham, 2006] Mike Worboys and Matt Duckham. Monitoring qualitative spatiotemporal change for geosensor networks. *International Journal of Geographical Information Science*, 20(10):1087–1108, November 2006.