

EMPIRICAL EVALUATION OF THE EXCLUSION APPROACH TO ESTIMATING CAMERA OVERLAP

Rhys Hill, Anton van den Hengel, Anthony Dick, Alex Cichowski, Henry Detmold

{rhys,hengel,ard,alex,henry}@cs.adelaide.edu.au
The Australian Centre for Visual Technologies
School of Computer Science
The University of Adelaide

ABSTRACT

Making intelligent decisions on the basis of the video captured by a large network of surveillance cameras requires the ability to identify overlap between their fields of view. Without this information it is impossible to perform even simple analysis, such as distinguishing between repeated behaviours and multiple views of the same behaviour. Large-scale intelligent video surveillance thus requires a means of understanding the relationships between the fields of view of the cameras involved. The exclusion approach is the only method currently capable of performing online estimation of camera overlap for networks of more than 50 cameras, with a version of the algorithm applicable to 1000 camera networks having been published. Empirical evaluation of every such algorithm is critical to assessing its performance, and essential if comparisons between methods are to be made. This paper presents a method by which such an empirical evaluation may be carried out, and makes publicly available the data (including ground truth) on which it is based in order that competing methods might be compared equally. Precision vs recall curves are reported for a series of experiments comparing the results of exclusion to ground truth. These results demonstrate the strengths and limitations of the exclusion-based estimation process, but show that the performance of the method exceeds the requirements of surveillance applications.

1. INTRODUCTION

Video surveillance networks serve a number of purposes including public safety, crime deterrence and perimeter security. The hardware and networking infrastructure required to support these networks is becoming increasingly widespread, leading to a proliferation of networked surveillance devices that constantly capture and store video data. The question then arises: what can one do with all this data? For a small network of, say, less than 10 cameras, it is possible (but expensive) to employ human operators to monitor the data as

it is captured, or search through archives for a specific event. An alternative is to use software to monitor incoming footage automatically, or to index and search video archives. Recently developed computer vision algorithms show some promise in this area, but are typically only applicable to small scale networks, and require validation (see [1] for a survey).

Installations of 50,000 camera surveillance networks are now being reported, and networks of more than 100 cameras are common place. Human operators struggle to concentrate on even a single video feed for any length of time, and even with sufficient manpower the problem of coordinating observations that span multiple cameras cannot be overcome manually. Computer vision algorithms require some idea of the arrangement of the cameras, either spatially or in terms of observed activity, in order to reason about events that span multiple cameras. Acquiring this information automatically typically involves accumulating evidence for links between camera pairs over time—a task that grows exponentially as cameras are added to a network.

The spatial and temporal relationships between the fields of view of a set of cameras can be described in terms of their *activity topology* [2]. An accurate estimate of activity topology has a number of uses in video surveillance. The most important of which is as a generic tool facilitating multi-camera algorithms through the prediction of the movement of targets through and between camera fields of view, and as a means of partitioning surveillance computations for scalable distributed processing.

The suitability of the *exclusion*-based approach [2,3] to the estimation of activity topology of networks of over 1000 cameras has been shown [4,5]. However a quantitative analysis of the accuracy of the method has not previously been provided, due to the difficulty of obtaining ground truth against which to measure and the lack of comparable methods.

The main contribution of this paper is that it reports quantitative evaluation of the accuracy of exclusion; based on measurements obtained from real video footage. Previous work has focused on validation of exclusion, by inspection and using simulated inputs, and on the performance properties of the

This work was supported by ARC Discovery Grant DP0770482 and the Government of South Australia PSRF scheme.

approach, most particularly scalability. A second major contribution is that we explore how to approach empirical evaluation of activity topology estimators in relation to ground truth topology, overcoming a number of significant challenges involved in undertaking such experimentation. Finally, to facilitate a more standardised approach to topology estimator evaluation, we are making part of our data set (including ground truth) freely available on the Internet.¹

2. MOTIVATION – ACTIVITY TOPOLOGY

An estimate of the activity topology associated with a camera network makes feasible a number of processes critical within on-line video surveillance. Nodes within the activity topology graph represent the fields of view of individual cameras, or alternatively regions within those fields of view. Each such region is labelled a *cell* and denoted c_x . The edges of the graph represent the connections between cells. These connections may be used to represent the overlap of the cells or, by including time offsets, the movement of targets through the graph. We are interested here only in assessing the ability of the exclusion algorithm to estimate the overlap in the fields of view of the cameras, and use the links in the graph to represent this information.

2.1. Formulation of Activity Topology

In order to focus on the special case (camera overlap) of interest in this paper, the definition of the activity topology graph is further elaborated as follows:

1. Edges are directed, such that (c_i, c_j) represents the flow from c_i to c_j whereas (c_j, c_i) represents the (distinct) flow from c_j to c_i . Directed edges can be converted to undirected edges if required, but the exclusion algorithm estimates each direction independently and thus we retain this information.
2. Each edge has a set of labels, $p_{i,j}^{[a,b]}$ for various time delay intervals $[a, b]$, each giving the probability that activity leaving c_i arrives at c_j after a delay between a and b . In this paper, each edge has exactly one such label, that for $[-\epsilon, \epsilon]$ where ϵ is some small value large enough to account for time differences between cameras. Thus $p_{i,j}^{[-\epsilon, \epsilon]}$ describes overlap between cameras.

Actual activity topologies are constrained by building layout, camera placement and other factors. Typical topologies contain sub-graphs with many edges between the nodes within the same sub-graph and few edges between nodes within different sub-graphs. These nearly isolated cliques are termed *zones* within the activity topology. Figure 1 shows a recovered activity topology for a network of over a hundred cameras, with zones represented by circles.

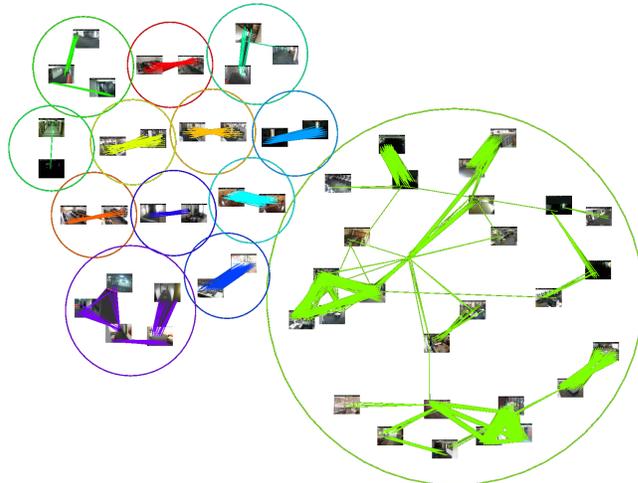


Fig. 1. Estimated activity topology for a real camera network. Edges linking cameras are shown as coloured lines, while zones are pictured as circles. Groups of size one have been omitted.

2.2. Applications of Activity Topology

An accurate estimate of activity topology has a number of applications within surveillance. The most important are *camera handover* and *activity partitioning*:

- *Camera handover* – for functions such as inter-camera tracking, when a target leaves a given camera it is necessary to continue the tracking in another camera. In the absence of any extra information, all other cameras must be searched to re-acquire the target. Activity topology assists in pruning this search space: only those cameras adjacent to the camera the target left (or even the cell it left from) within the topology need be searched. Further, when topology edges are labelled with likelihood, these likelihood estimates can be used to prioritise the order in which adjacent cameras should be searched. Finally, if topology edges are labelled with time-related properties, these may be used to determine when it is most propitious to start searching within each adjacent camera. Tracking is the most important use of this information, but it can be used in other contexts as well, and thus activity topology supports a generic “next cell” predictive function.
- *Activity partitioning* – activity topology provides a basis for partitioning distributed surveillance processing such that communication between partitions (which may be expensive) is minimal and the vast majority of communication is within partitions (which is inexpensive). This partitioning scheme enables an efficient divide-and-conquer approach to many surveillance functions.

¹Data available at <http://www.acvt.com.au/research/surveillance/>

2.3. Ground truth for activity topology

Without a quantitative analysis of the accuracy of activity topology estimation, it is impossible to gauge or predict the accuracy of the many surveillance functions that may be built on top of it. We therefore see this analysis as a process that must be undertaken to create reliable, well understood large scale surveillance software.

At the time of writing, the ground truth data that is required for such analysis is scarce. The PETS [6] workshop has a tradition of providing a standard data set for attendees to test their algorithms; however this is limited to data from a very small number of cameras, often only one. Existing large camera networks are not set up for evaluation, instead being laid out according to security demands. The sheer volume of data required is also an obstacle, both in terms of the raw number of bytes of video but also the amount of activity that must be observed to realistically obtain an estimate of topology. Additionally, privacy concerns prevent the use of data from many surveillance networks in public places.

3. PRIOR WORK ON ESTIMATING ACTIVITY TOPOLOGY

A number of existing surveillance systems rely on human operators to input the activity topology. Practical experience indicates this is unreliable: people generally have a poor ability to discern activity topology from a given camera configuration. This is in part due to the difficulty of recording the broad spatial relationships between cameras with the accuracy required, but also because such relationships only partly determine activity topology, as they do not account for autonomy in the behaviour of people and other objects under surveillance.

Previously, activity topology has been learnt by tracking people as they appear and disappear from camera fields of view (FOVs) over a long period of time. For example, in [7] the delay between the disappearance of each person from one camera and their appearance in another is stored to form a set of histograms describing the transit time between each camera pair. The system is demonstrated on a network of 3 cameras, but does not scale easily as it requires that correspondences between tracks are given during the training phase when topology is learnt.

Dick et al. [8] suggest an alternative approach whereby activity topology is represented by a Markov model. This does not require correspondences, but does need to learn a dense $N \times N$ transition matrix during the training phase and so does not scale well with the number of cameras N , due to the number of observations required for the Markov model.

The training phase required in this previous work is problematic in large networks, chiefly because the camera configuration, and thus activity topology, changes with surprising frequency; as cameras are added, removed, moved and fail.

Approaches requiring a training phase to complete before operation would have to cease operation each time there is a change, and only resume once re-training has completed. This is an intolerable restriction on the availability of a surveillance network.

Hence, on-line automatic approaches, where topology is estimated concurrently with the operation of surveillance, are desirable. Ellis et al. [9] do not require correspondences or a training phase, instead observing motion over a long period of time and accumulating appearance/disappearance information in a histogram. Instead of recording known correspondences, it records every possible disappearance that could be related to an appearance. Over time, actual transitions are reinforced and can be extracted from the histogram with a threshold. A variation on this approach is presented in [10], and has been extended by Stauffer [11] and Tieu et al. [12] to include a more rigorous definition of a transition based on statistical significance, and by Gilbert et al. [13] to incorporate a coarse to fine topology estimation. These methods rely on correctly analysing enough data to distinguish true correspondences, and have only been demonstrated on networks of less than 10 cameras.

4. ESTIMATING ACTIVITY TOPOLOGY BY EXCLUSION

Consider the problem of determining overlap for a set of N cameras. The set of cameras generates N images at time t , with each image partitioned into a grid of cells. Application of foreground detection [14] to all camera images produces a set of foreground blobs, each of which can be summarised into a position given by a single cell within the containing camera. At any given time t , each cell is labelled *occupied* or *unoccupied* depending on whether it contains a summarised foreground object.

Exclusion is based on the observation that a cell which is *occupied* at time t cannot be an image of the same area as any other cell that is simultaneously *unoccupied*. Given that cells tend to be unoccupied more often than they are occupied, this observation can be used to eliminate a large number of cell pairs as potentially viewing the same area at each time instant. The process of elimination can be repeated for each frame of video to rapidly reduce the number of pairs of image cells that could possibly overlap. This is the opposite of most previous approaches: rather than accumulate positive information about overlap between cells, we seek negative information allowing the instant elimination of impossible overlaps. Such overlaps are referred to as having been *excluded* [2].

In this paper, we only evaluate the application of exclusion to detecting overlap. Note however, that the technique is not limited to this special case of activity topology, but can also be applied to the general case (connections between non-overlapping cameras) through the use of varying time offsets in the operands to the exclusion operation. Future papers will

evaluate this scenario.

4.1. Exclusion over multiple timesteps

Rather than calculate exclusion separately at each timestep, it is more efficient to gather occupancy information over multiple frames and then calculate exclusion over all of them at once.

Let the set of cells over all cameras be $\mathcal{C} = \{c_1 \dots c_k\}$. Corresponding to each cell c_i is an occupancy vector $\mathbf{o}_i = (o_{i1}, \dots, o_{iT})'$ with o_{it} set to 1 if cell c_i is occupied at time t , and 0 if not. If two cells are images of exactly the same region in the world, we would expect their corresponding occupancy vectors to match exactly. This can be tested by applying the *exclusive-or* operator \oplus to elements of the occupancy vectors,

$$\mathbf{o}_i \oplus \mathbf{o}_j = \max_{t=1}^T o_{it} \oplus o_{jt}. \quad (1)$$

It can be inferred that two cells c_i and c_j do not overlap if $\mathbf{o}_i \oplus \mathbf{o}_j = 1$. This comparison is very fast to compute, even for long vectors.

4.2. Tolerance for sources of error

Exclusion as described so far assumes that:

1. corresponding cells in overlapping cameras cover exactly the same visible area in the scene,
2. all cameras are synchronised, so they capture frames at exactly the same time,
3. the foreground detection module never produces false positives or false negatives, and
4. the ground truth does not change over time.

In reality none of these assumptions is likely to hold completely. It is thus possible that two overlapping cells might simultaneously register as occupied and vacant and therefore that the exclusive-or of the corresponding occupancy vectors might incorrectly indicate that they do not overlap.

Assumption 1 can be relaxed by including the neighbours of a particular cell when registering its occupancy. We use a padded occupancy vector \mathbf{p}_i which has element p_{it} set to 1 when cell c_i or any of its eight-connected neighbours are occupied at time t . A more robust mechanism for determining whether two cells c_i and c_j overlap is thus to calculate $\mathbf{o}_i \ominus \mathbf{p}_j$ on the basis of the occupancy vector \mathbf{o}_i and the padded occupancy vector \mathbf{p}_j . The \ominus operator is a uni-directional version of the exclusive-or defined such that

$$\mathbf{o}_i \ominus \mathbf{o}_j = \max_{t=1}^T o_{it} \ominus o_{jt}, \quad (2)$$

where $o_{it} \ominus o_{jt}$ is 1 if and only if o_{it} is 1 and o_{jt} is 0. Note that this exclusion calculation is no longer symmetric. Also, padded occupancy is properly defined only for cells with

neighbours on all sides, with the effect that camera resolution (in terms of numbers of cells) is reduced by two in each dimension.

The *spatial padding* approach just described will tend to overcome the effects of clock skew (assumption 2), provided that skew is also minimised through keeping camera clocks closely synchronised (via NTP). In addition, *temporal padding* is used, whereby \mathbf{p}_i has element p_{it} set to 1 when cell c_i or any of its neighbours is occupied at any time in the range $t \pm \epsilon$.

To account for detection errors (assumption 3), we calculate exclusion based on accumulated results over multiple tests, rather than relying on a single contradictory observation. Assuming that the detector has a constant failure rate, the evidence for exclusion is directly related to the number of contradictory observations in a fixed time period $t = 1 \dots T$, which we call the *exclusion count*:

$$E_{ij} = \sum_{t=1}^T o_{it} \ominus p_{jt}. \quad (3)$$

Finally it is possible for the ground truth to change over time (violating assumption 4). The experiments we describe in this paper have constant ground truth to prevent this being a source of error. Note, however, that we hypothesise that exclusion is well suited to providing estimation of changing overlap, and future work will explore this hypothesis.

4.3. Normalised exclusion

The exclusion count has two main shortcomings as a measure for deciding cell overlap/non-overlap:

- As the operator $a \ominus b$ will only return true when a is true, the exclusion count E_{ij} between cells c_i and c_j is bounded by the number of detections in c_i , and is likely to be higher for cells c_i that register more detections.
- In a large network, it will frequently occur that data sent from a camera will be lost, or not arrive in time to be included in the exclusion calculation, or that a camera will go offline. Thus the maximum value of E_{ij} also depends on how often data from c_j is available.

To address these problems we define a padded *availability* vector \mathbf{v} for each cell that is set to 1 when occupancy data for the cell and its neighbours is available, and 0 otherwise. We can then define an exclusion *opportunity count*,

$$O_{ij} = \sum_{t=1}^T o_{it} \wedge v_{jt}, \quad (4)$$

between each pair of cells. Based on this we define an overlap certainty measure from each cell with opportunity count at least 1 to every other cell

$$C_{ij} = \frac{O_{ij} - E_{ij}}{O_{ij}}. \quad (5)$$

This measures the number of times that an exclusion was not found between c_i and c_j as a proportion of the number of times an exclusion could possibly have been found given the available data.

4.4. Deriving overlap from exclusion

Intuitively, overlap is symmetric: if c_i overlaps with c_j , then c_j overlaps with c_i . This is utilised to strengthen estimation of overlap from exclusion certainty. Overlap of c_i and c_j is estimated by the following Boolean function,

$$X_{ij} = C_{ij} > C^* \wedge C_{ji} > C^*, \quad (6)$$

with C^* a threshold value.

5. EVALUATION APPROACH

We aim to evaluate the accuracy of exclusion in estimating camera overlap. Our approach is to measure error in the estimates produced by exclusion when compared with the ground truth overlap, including both false positives (overlap in the estimate but not in ground truth) and false negatives (overlap in ground truth but not in the estimate).

Accuracy is evaluated in terms of precision-recall [15] curves, which account for both false positive and false negative errors. In this approach, the relationship between estimation (the overlap results produced by exclusion) and reality (the ground truth overlap) is expressed in terms of a *confusion matrix*.

	ground truth	
	overlap	no overlap
estimated overlap	<i>TP</i>	<i>FP</i>
no estimate	<i>NEP</i>	<i>NEN</i>
estimated no overlap	<i>FN</i>	<i>TN</i>

Table 1. Confusion Matrix

where *TP* (true positives) is the number of cell pairs correctly estimated to overlap, *FP* (false positives) is the number of cell pairs incorrectly estimated to overlap, *NEP* (no estimate positives) is the number of overlapping cell pairs for which there is insufficient data to produce an estimate, *NEN*, (no estimate negative) is the number of non-overlapping cell pairs for which there is insufficient data to produce an estimate, *FN* (false negatives) is the number of cell pairs incorrectly estimated not to overlap and *TN* (true negatives) is the number of cell pairs correctly estimated not to overlap. We define

$$Recall = \frac{TP}{TP + FN}, \quad (7)$$

and

$$Precision = \frac{TP}{TP + FP}. \quad (8)$$

Notice we remove the cases where there is insufficient occupancy data for the exclusion based estimator to draw a conclusion. In the experiments described in this paper, we use a fixed cut-off, and any cell pairs (c_i, c_j) for which

$$O_{ij} < O^* \quad \text{or} \quad O_{ji} < O^* \quad (9)$$

holds are placed into the *NEP* and *NEN* categories. We choose a fixed low sample-size cut-off, $O^* = 20$, in order to avoid interference with the varying threshold, C^* , evaluated in precision–recall curves. This is a pragmatic choice made in the experimental design; our other work on exclusion uses a log based penalty term.

Now, *Recall* measures the fraction of ground truth overlapping cell pairs that are correctly identified by the estimator, whereas *Precision* measures the cell pairs correctly identified as overlapping by the estimator as a fraction of all cell pairs identified as overlapping by the estimator. A threshold, C^* , is used by the estimator to determine whether a cell pair is considered to overlap. Raising C^* converts false positives into true negatives, hence increasing *Precision* but also converts true positives into false negatives, reducing *Recall*. The precision–recall curve captures the effect of varying the threshold.

6. EXPERIMENTS AND RESULTS

In all experiments, each camera’s field of view is divided into a 12 by 9 grid of cells with edge cells omitted; there are thus 70 cells per camera. The number of cells per field of view can be adjusted to suit the application, but it has been determined empirically that 12 by 9 is a sufficient to accurately represent the relationships required. The selection thus partly reflects the density of the ground truth correspondences. Cameras operate at a frame rate of either 10 FPS or 30 FPS and at 640x480 resolution. The time padding, ϵ , is set to 0.15 seconds to account for the differing camera frame rates. As mentioned previously, the low sample-size cut-off O^* is set to 20. The precision-recall curves reported for each experiment show the effect of varying the value of C^* between 0 and 1, in steps of 0.05. Finally, the minimum blob size threshold for identifying foreground objects is set to 200 pixels.

6.1. Measuring and matching ground truth

The ground truth data presented in this paper was captured by five cameras, each mounted on a tripod and looking towards the floor. The views were chosen to provide a significant level of overlap between pairs of cameras. Markers were placed on the floor at each corner of each view, colour coded to allow easy inspection. The locations of these markers were also measured within the view of each camera, to provide a mapping between the camera’s image co–ordinates and floor co–ordinates.

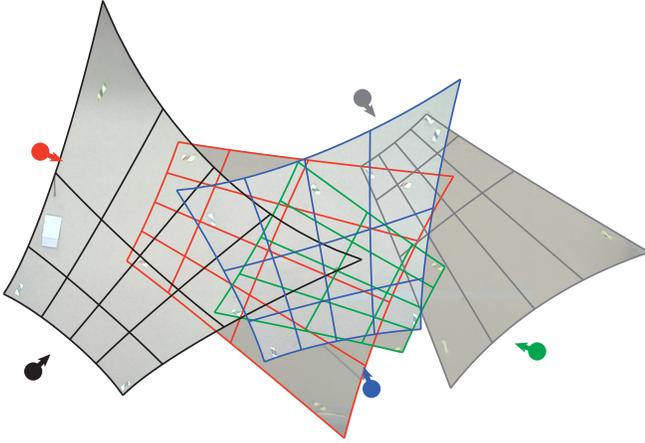


Fig. 2. Ground truth overlap for five cameras as used in several of our experiments. Only every third grid line is shown.

Surveillance cameras often use small, cheap lenses which exhibit a significant radial distortion. Therefore, each camera was examined and the degree of radial distortion measured and used to correct the images. The mapping between the image and floor co-ordinates can then be used to align the images, shown in Figure 2, allowing the overlap of each camera cell to be computed. For clarity, the figure shows cells at 3 by 4 resolution instead of 12 by 9, so each cell in the figure represents a 3 by 3 grid of cells.

The topology computed in this way has some limitations. Any degree of overlap between floor regions is considered valid and forms a link in the graph. In accepting such small levels of overlap, it is assumed that the accuracy of the floor mapping is very high, however in practice it is difficult to know exactly how accurate it is – the alignment is based on image data with finite resolution, which has been transformed in a non-linear way onto the floor.

The topology algorithm itself is immune to the effects of radial distortion and other imaging issues, since it operates on observations only, and does not use or need any geometric information. Overly stringent matching to the ground truth can thus produce pessimistic results, which are not indicative of the true accuracy of the method. For this reason, two methods for comparing against the ground truth will be presented. The first is a simple difference between the two graphs, the second will allow a one cell difference in either the source or destination cell to be considered a match. These matching rules are related: the former has a tolerance of zero cells, and the latter a tolerance of one. The true accuracy of the method is between these two points. Note that any link which has an error greater than the tolerance will still be reported, regardless of any correct neighbouring links.

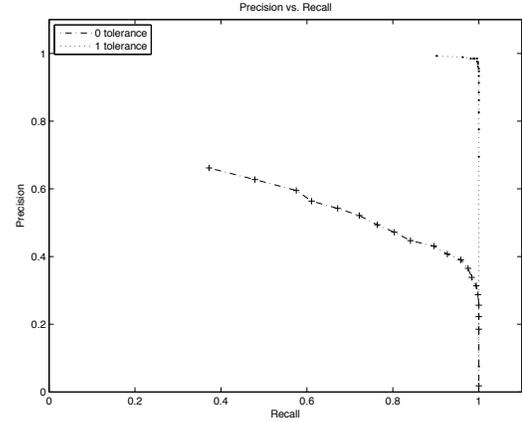


Fig. 3. Precision-recall curves for model cars driving data

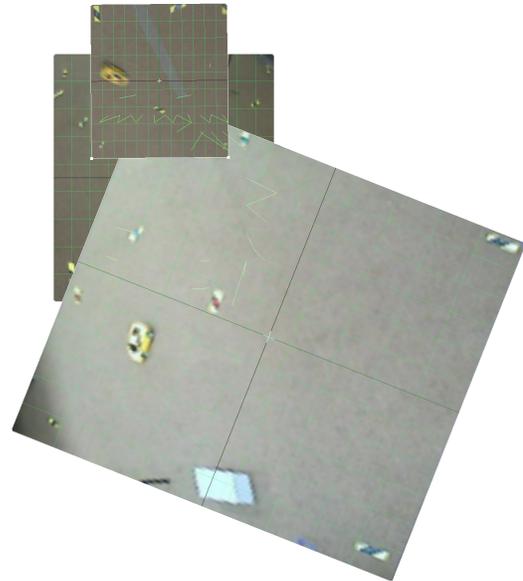


Fig. 4. Erroneous links remaining with $C^* = 0.95$

6.2. Planar cells and activity

In this experiment, occupancy data is generated by activity (two remote controlled cars) moving on the floor, for thirty minutes. As these objects are effectively on the ground plane, any sensitivity to the minimum visible extent calculation is eliminated. The aim of this experiment is to evaluate the accuracy of exclusion under ideal conditions.

Precision-recall results for this experiment are shown in Figure 3. The two curves are for a ground truth matching tolerance of zero and one cells respectively. Figure 4 shows an illustrative subset of false positive overlaps with $C^* = 0.95$ and the use of zero cell tolerance matching. These are off by one cell from the ground truth and hence counted true positives by the one cell tolerance rule. This effect explains why *Precision* approaches 1 with the one cell tolerance.

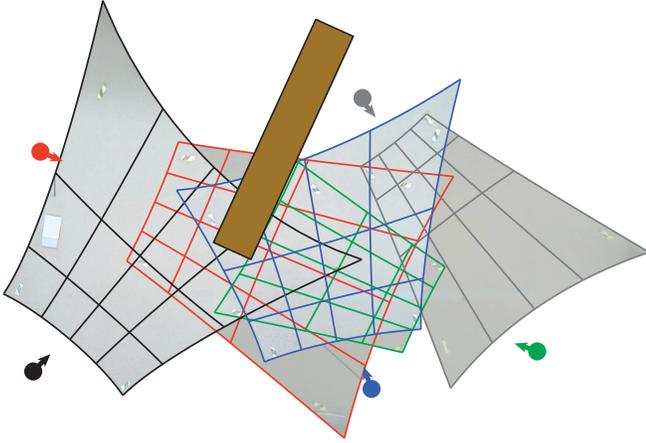


Fig. 5. Introducing an occlusion into the scene

Consider the use of the estimated topology in tracking (*i.e.* for camera handover). Starting from a given current cell, the set of cells to be searched to continue tracking is termed the *candidate set*. It is critical that the candidate set includes the actual next cell, in order to avoid lost tracks, thus the minimal candidate set implied by a given ground truth overlap (there might be several overlaps for a given starting cell) includes the overlapping cell and its immediate neighbours, to tolerate movement between frames. Now, consider the one cell tolerance results (with $Recall = 1, Precision = 1$), which imply that a candidate set including the cell estimated to overlap, its neighbours, and the neighbours of those neighbours includes the actual next cell. Thus the size of the candidate set derived from this technique is at most 25 cells per ground truth overlap, which provides an entirely acceptable search space.

6.3. Planar cells, planar activity, occlusion

This experiment differs from 6.2 only by the introduction of an occlusion into the scene, namely a bookshelf that is sufficiently tall that no camera can see over the top of it. There are no other changes from 6.2, but of course the occlusion will perturb the ground truth. The aim of this experiment is to evaluate the accuracy of exclusion with a more realistic scene than in 6.2. This test was performed on less video data than the others, so the low sample-size cut-off O^* was reduced to 8 to take this into account. The results presented for this experiment are the links which have been missing from the estimated topology, shown in Figure 6. As expected, the missing links are those which cross the book-shelf; all others are present.

6.4. Planar cells, non-planar activity

This experiment has the same camera configuration as 6.2, the only difference is that the activity generating occupancy data is the movement of three dimensional objects (people)

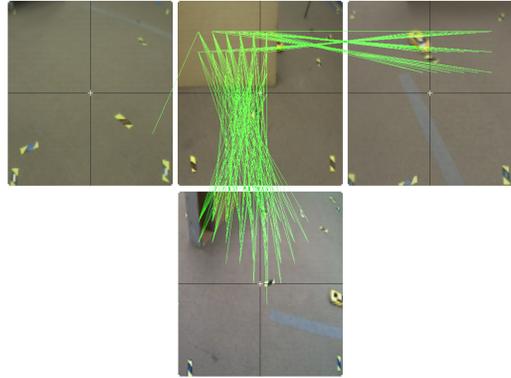


Fig. 6. Links which are missed when an occlusion is introduced into the scene.

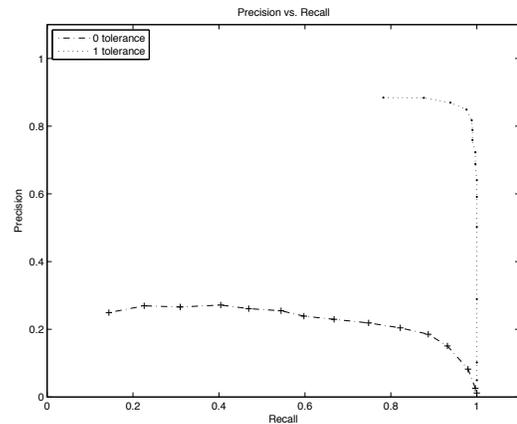


Fig. 7. Precision-recall curves for people walking data

in the space above the ground plane. The effect is that the minimum visible extent calculation becomes significant, and a potential additional source of error. Note that the ground truth for actual overlap of fields-of-view on the floor plane is the same as in experiment 6.2. The aim for this experiment is to evaluate the accuracy of exclusion with more realistic activity than in 6.2.

The precision-recall curves in this case show poor accuracy with the zero cell tolerance matching rule, whereas the curve for one cell tolerance is still excellent. It can be concluded from these results that the estimates produced by exclusion are less accurate with realistic activity, which is to be expected. Legs are difficult to measure from closely spaced views, since from one angle they can appear widely separated, whilst from another they will appear to touch, leading to erroneous measurements.

However, consider the tracking application in relation to the one cell tolerance results. With ($Recall = 1, Precision = 0.8$), there are $1.25 = 1/0.8$ estimated overlaps per true overlap. As a result, the set of candidate cells for continuing tracking from a given cell will contain an average of

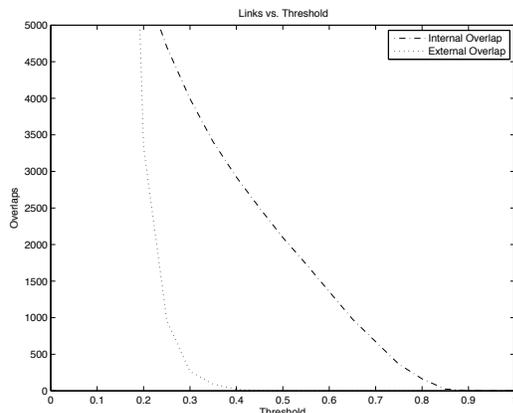


Fig. 8. Interference from non-overlapping cameras

31.25 = (1.25 × 25) cells for each true overlap, again an acceptable search space.

6.5. A zone within a large topology

In this experiment we execute exclusion for a moderately large network (approximately 100 cameras) within which the five cameras from 6.3 form an isolated sub-graph (a strong form of zone) within the overall topology. Our aims are as follows:

- To verify that the topology computed for the sub-graph is unaffected by the rest of the graph.
- To measure the false overlap between cells within the sub-graph and those outside. Note that there is no actual overlap, thus any overlap estimated by exclusion is erroneous.

In addition we note in passing that our distributed implementation of exclusion processes the data of this experiment in better than realtime; readers interested in such performance aspects are referred to [4].

Results for this experiment are shown in Figure 8. The external overlap shows the subset of false positives relating to (false) overlap between cells within the zone and cell outside it (*i.e.* interference from outside the zone). The internal overlap shows the total positives for overlap within the zone. Both are plotted against C^* threshold values. It is evident that relatively low C^* values (≥ 0.4) are sufficient to eliminate interference completely. Such C^* values obtain acceptable precision and recall (0.72 and 0.99 respectively).

7. CONCLUSION

This paper has presented an analysis of the accuracy of the exclusion algorithm, based on real data. Through a series of experiments, it has been shown that the exclusion algorithm

yields very accurate results, more than acceptable to provide a basis for other algorithms, such as tracking or behaviour analysis across large networks of cameras. These tests have been carried on a small test network with a simple geometry, due to the difficulty involved in manually generating ground truth data. Other tests show that the method scales to larger networks, and the combination of these two sets of tests demonstrates both the applicability of the method to the problem of topology estimation for large camera networks. Comparisons to other methods have not been presented because, as far as we are aware, no other methods capable of estimating the topology of large networks exist.

A carefully captured and calibrated set of ground truth data has been presented – in summary form in this document, with video data being available online. Such data will allow direct comparison of different topology estimation algorithms, critical for advancement of the field.

The exclusion algorithm has shown itself to be highly accurate and scalable, both critical features for current and future large surveillance camera networks.

8. REFERENCES

- [1] M. Valera Espina and S. A. Velastin, “Intelligent distributed surveillance systems: A review,” *IEE Proceedings - Vision, Image and Signal Processing*, vol. 152, no. 2, pp. 192–204, April 2005.
- [2] A. van den Hengel, A. R. Dick, and R. Hill, “Activity topology estimation for large networks of cameras,” in *Proceedings of the IEEE International Conference on Advanced Video and Signal-based Surveillance*. November 2006, IEEE.
- [3] Anton van den Hengel, Anthony R. Dick, Henry Detmold, Alex Cichowski, and Rhys Hill, “Finding camera overlap in large surveillance networks,” in *8th Asian Conference on Computer Vision*, 2007, pp. 375–384.
- [4] H. Detmold, A. van den Hengel, A. R. Dick, A. Cichowski, R. Hill, E. Kocadag, K. Falkner, and D. S. Munro, “Topology estimation for thousand-camera surveillance networks,” in *Proceedings of First ACM/IEEE International Conference on Distributed Smart Cameras*. September 2007, pp. 195–202, IEEE.
- [5] Henry Detmold, Anton van den Hengel, Anthony Dick, Katrina Falkner, David S. Munro, and Ron Morrison, “Middleware for distributed video surveillance,” *IEEE Distributed Systems Online*, vol. 9, no. 2, 2008.
- [6] James Ferryman, Ed., *10th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, <http://www.pets2007.net/>, IEEE, 2007.
- [7] O. Javed, Z. Rasheed, K. Shafique, and M. Shah, “Tracking across multiple cameras with disjoint views,” in *IEEE Int. Conf. Computer Vision*, 2003, pp. 952–957.
- [8] A.R. Dick and M. J. Brooks, “A stochastic approach to tracking objects across multiple cameras,” in *Proc. Australian Joint Conference on Artificial Intelligence*, 2004, pp. 160–170.

- [9] T. J. Ellis, D. Makris, and J.K. Black, "Learning a multi-camera topology," in *Joint IEEE Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, 2003, pp. 165–171.
- [10] T. H. Ko and N. M. Berry, "On scaling distributed low-power wireless image sensors," in *Proc. 39th Annual Hawaii International Conference on System Sciences*, 2006, vol. 09, p. 235.
- [11] C. Stauffer, "Learning to track objects through unobserved regions," in *IEEE Computer Society Workshop on Motion and Video Computing*, 2005, pp. II: 96–102.
- [12] K. Tieu, G. Dalley, and W.E.L. Grimson, "Inference of non-overlapping camera network topology by measuring statistical dependence," in *Proc. IEEE International Conference on Computer Vision*, 2005, pp. II: 1842–1849.
- [13] A. Gilbert and R. Bowden, "Tracking objects across cameras by incrementally learning inter-camera colour calibration and patterns of activity," in *European Conference on Computer Vision*, 2006, vol. 2, pp. 125–136.
- [14] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.
- [15] Vijay Raghavan, Peter Bollmann, and Gwang S. Jung, "A critical investigation of recall and precision as measures of retrieval system performance," *ACM Trans. Inf. Syst.*, vol. 7, no. 3, pp. 205–229, 1989.