CAVIAR

Context Aware Vision using Image-based Active Recognition

IST- 2001- 37540

D23

Report on top-down primed salience mechanisms

CAVIAR :	D23
Date :	September 30, 2005
Author(s) :	Fisher, Hall, Vasquez
Work package :	3
Document status :	Version 1.0
Usage :	public
Keywords :	model selection, target selection

1 Introduction

The issue being considered in this deliverable is how contextual knowledge could be used to focus attention and processing. This focussing gives the advantages of: a) more efficient processing by reducing the number of targets and hypotheses considered and b) more correct processing by reducing the potential for erroneous targets and hypotheses.

Two processes were investigated and are reported here:

1. **Model Selection:** When recognising behaviour, there are up to 10 different models that can be applied to each of as many as 10 different targets in each video sequence frame. It is possible to exhaustively try each model for each target. But it is also possible to prioritise the models for each target. Then, the recognition process can work through the priority list for as long as it has resources.

   Section 2 describes a process that uses a Bayes' network to integrate information from 4 sources: behaviour importance priorities, incompatibility between model and data, previous frame results and scene position. The resulting process ranks the true model (as defined in the ground truth) in average position 1.2, where ranks positions are defined as 1, 2, ... N if N models are possible.

2. **Tracked Target Focussing:** When applying a change-based target detector, the question is where to apply this process. Processing the full image should find all targets, but is much slower and might also allow more spurious targets arising from noise (which can occur anywhere in the image as well as in the detection zones).

   Section 3 presents experiments with an entry detection zone scheme, which exploits *a priori* scene knowledge that lets us know where targets are likely to appear from. Targets that somehow appear in the middle of the scene would then not be detected, but this is rather unlikely. The experiments consider: where to place detection zones (systematically instead of randomly) and how often to check them (about once every 4 frames) before the true and false detection rates are significantly affected. Initial experiments also suggest that the detection strategy can be learned.

These two techniques show promise for the future, but were not implemented in the demonstrator, because there we did not consider execution rate issues.

2 Model Selection

The current filtering module for the Caviar project is based on a Bayesian Model Network. The module generates two output values for every object on the scene: a hypotheses list and a weight value (Interestingness evaluation). The Bayesian network output is an XML file that contains a list with all the possible hypotheses for every object on the scene and their associated weight (according to their interestingness).

Since the purpose of the model is to serve as a filter for the Context Generator Module, the final output of the module should be a prioritised list of pairs (person, hypothesis). The priority

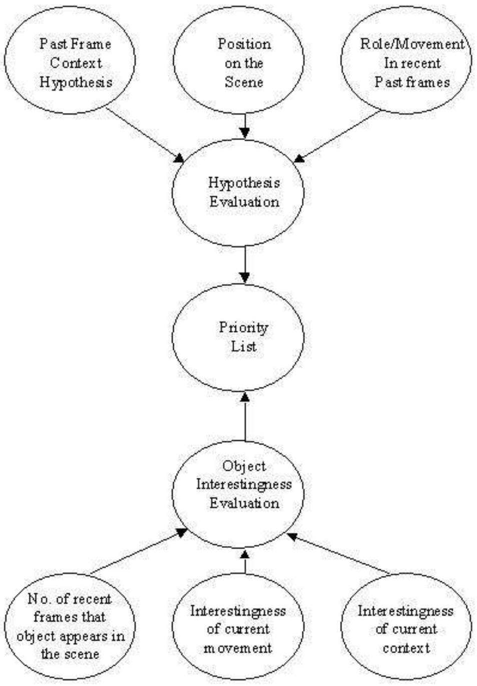
Past Frame Context Hypothesis
Position on the Scene
Role/Movement In recent Past frames
Hypothesis Evaluation
Priority List
Object Interestingness Evaluation
No. of recent frames that object appears in the scene
Interestingness of current movement
Interestingness of current context

Figure 1: Bayesian Network for the priority list

list is defined as a descending order list of pairs (person, hypothesis). The value that determines the sorting order is the product of the associated weight for the person and the hypothesis probability. According to the current XML Schema the list is created by the module that receives the XML output file from the filtering module. An example of the current output for the module is depicted below:

```
<dataset name="">
    <frame number="0">
        <objectlist>
            <object id="2">
                <orientation>174.000000</orientation>
                <box x="89.000000" y="202.000000" w="40.000000" h="16.000000" />
                <appearance>visible</appearance>
                <hypothesislist>
                    <hypothesis id="0" prev="0" evaluation="0.307692">
                        <context evaluation="1.000000">browsing</context>
                    </hypothesis>
                    <hypothesis id="1" prev="0" evaluation="0.153846">
                        <context evaluation="1.000000">drop down</context>
                    </hypothesis>
                    <hypothesis id="2" prev="0" evaluation="0.153846">
                        <context evaluation="1.000000">windowshop</context>
                    </hypothesis>
                    <hypothesis id="3" prev="0" evaluation="0.076923">
                        <context evaluation="1.000000">walking</context>
                    </hypothesis>
                    <hypothesis id="4" prev="0" evaluation="0.076923">
                        <context evaluation="1.000000">shop enter</context>
                    </hypothesis>
                    <hypothesis id="5" prev="0" evaluation="0.076923">
                        <context evaluation="1.000000">shop exit</context>
                    </hypothesis>
                    <hypothesis id="6" prev="0" evaluation="0.076923">
                        <context evaluation="1.000000">shop reenter</context>
                    </hypothesis>
                    <hypothesis id="7" prev="0" evaluation="0.076923">
                        <context evaluation="1.000000">immobile</context>
                    </hypothesis>
                </hypothesislist>
                <featurelist>
                    <weight>1.500000</weight>
                </featurelist>
            </object>
        </objectlist>
    </frame>
</dataset>
```

2.1 Context Evaluation

The filter module produces a hypotheses list that contains all the possible hypothesis. The probability of a context for an object in the current scene is evaluated based on three factors:

1: The movement and role for the past *n* frames.

2: The position of the person in the current scene

3: The hypotheses inferred by the Hypotheses generator in the previous frame.

These three factors are combined using a Bayesian network approach. The formula used to determine the probability of a hypothesis for the current object is:

$$P(c_i^t) = \frac{P\left(c_i^t/(m,r)_{t-frames}^t\right) * P\left(c_i^t/pos^t\right) * P\left(c_i^t/C^{t-1}\right)}{\sum_{k=0}^{N} P\left(c_k^t/(m,r)_{t-frames}^t\right) * P\left(c_k^t/pos^t\right) * P\left(c_k^t/C^{t-1}\right)}$$

Where:

C = the set of possible contexts for the current object. For a person object the current set is:

C = {walking, shop reenter, shop enter, shop exit, window shop, immobile, browsing, drop down}

For a group of persons object the current set is:

C = {walking, shop reenter, shop enter, shop exit, window shop, browsing, meeting, fighting}

c_i^t = the *i*th context from the context set at frame *t*.

M = the movement value of the individual

R = the role value of the individual.

pos^t = the position of the individual in the current frame.

N = the total number of contexts.

The hypothesis probabilities are normalized, according to the current CAVIAR Inference Modules.

2.1.1 Hypothesis inference from past frame Hypotheses.

The conditional probability for a hypothesis in the current frame based on the hypotheses inferred by the Hypotheses Generator Module (David Tweed's Module) is determined as:

$$P(c_i^t/C^{t-1}) = \frac{P_1 + \sum_{j=0}^{N} P\left(c_i^t/c_j^{t-1}\right)}{N * P_1 + \sum_{i=0}^{N}\sum_{j=0}^{N} P\left(c_i^t/c_j^{t-1}\right)}$$

Where the conditional probability $P\left({c_i^t}/{c_j^{t-1}}\right)$ is defined as:

$$P\left({c_i^t}/{c_j^{t-1}}\right) = \begin{cases} P\left(c_j^{t-1}\right) & c_i^t = c_j^{t-1} \\ 0 & c_i^t \neq c_j^{t-1} \end{cases}$$

and P_1 is a parameter defined by the user. Large values of P_1 diminish the effect of the previous contexts on the priority list, while small values of P_1 increase the effect of the previous context on the list. $P\left(c_j^{t-1}\right)$ is obtained from the probabilistic recogniser (David Tweed's module).

2.1.2 Hypothesis inference from the position in the scene

Currently there are three locations for the sequences, the first one is a lobby in an office in France, the other two scenarios are in a commercial shopping centre in Portugal.

In the first scenario there are four zones that have been identified as browsing areas (Figure 2). In the Portugal scenarios there are some zones that have been identified as entrances to the shops (Figure 3 and 4).

Browsing areas are associated with contexts that contain a browsing situation in their definition according to the current CAVIAR model. In the current implementation for an individual c_{zones}=$\{$browsing$\}$, for a group c_{zones}=$\{$browsing$\}$.

Entrance areas are associated with contexts that contain shopping situations in their definition according to the current CAVIAR model. In the current implementation for an individual c_{zones}=$\{$shop enter, shop exit, shop reenter, window shop$\}$, for a group c_{zones}=$\{$shop enter, shop exit, shop reenter, window shop$\}$.

Outside of these zones the probability for all the hypotheses of an object will be equal $c_{zones} = c$ since the set of zone contexts contains the complete set of contexts.

The probability of a hypothesis given the current position of the object is:

$$P({c_i^t}/{pos^t}) = \begin{cases} P_{zone} & c_i^t \in c_{zones} \\ P_{out\_zone} & c_i^t \notin c_{zones} \end{cases}$$

$$P_{zone} = {w_{zone}}/{w_{total}}$$

$$P_{out\_zone} = {w_{out\_zone}}/{w_{total}}$$

$$w_{total} = n(c_{zones}) * w_{zone} + (N - n(c_{zones})) * w_{out\_zone}$$

Where:

$n(c_{zones})$: Total of suitable contexts for the current area according to the object position.

N: Total number of contexts in the current context set.

w_{zone}: Weight for hypotheses contained in the current c_{zones} set.

w_{out_zone}: Weight for hypotheses not contained in the current c_{zones} set.

w_{zone} and w_{out_zone} are user defined variables. Currently $w_{out\_zone} = 1$ and w_{zone} is a parameter defined by the user.

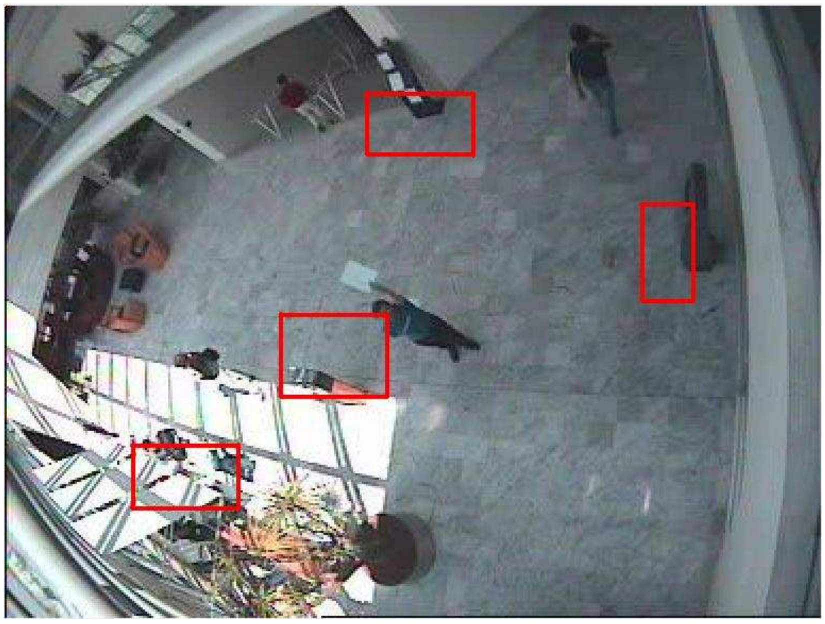

Figure 2: Browsing Areas for France Scenario

2.1.3 Hypothesis inference from roles and movement in previous frames.

The conditional probability for a hypothesis is inferred for an object based on roles and movements inferred by the CAVIAR inference module from the last f frames. In each past frame a variable number of hypotheses are generated for an object. The hypotheses contain the movement and sometimes contain the role. The formula to determined the probability is:

$$P\left(c_i^t/(m,r)^{t-f,t-1}\right) = \frac{P_2 + \sum_{j=t-f}^{t-1}\sum_{k=0}^{H} P\left(c_i^t/(m_k,r_k)^j\right)}{N * P_2 + \sum_{l=0}^{N}\sum_{j=t-f}^{t-1}\sum_{k=0}^{H} P\left(c_l^t/(m_k,r_k)^j\right)}$$

$$P\left(c_i^t/(m_k,r_k)^j\right) = P\left(c_i^j/(m_k,r_k)^j\right)$$

$$P\left(c_i^j/(m_k,r_k)^j\right) \equiv \begin{cases} P(m_k,r_k)^j & c_i^j \subset (m_k,r_k) \\ 0 & c_i^j \not\subset (m_k,r_k) \end{cases}$$

And:

$$P\left(c_i^j/(m_k,r_k)^j\right) \equiv \begin{cases} P(m_k,r_k)^j & c_i^j \subset (m_k,r_k) \\ 0 & c_i^j \not\subset (m_k,r_k) \end{cases}$$

Where:

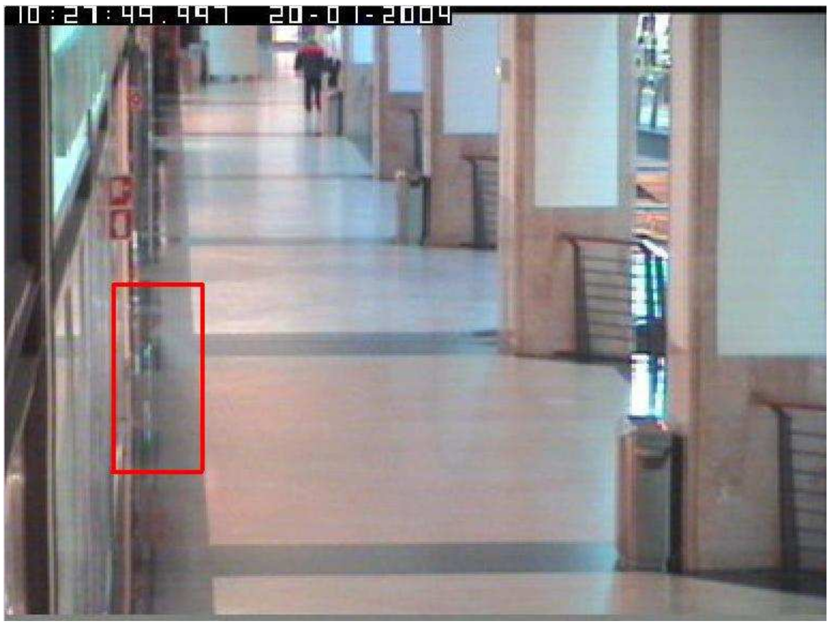

Figure 3: Entrance to Shop Areas for Portugal shopping centre (corridor view)

$P\left(c_i^t / (m_k, r_k)^j\right)$: Probability for context *i*at time *t* given the movement m_k and role r_k at time *j*.

H: Total number of hypotheses at the current frame.

$c_i^j \subset (m_k, r_k)$: context *i* contains a situation for the movement and role *k* according to the current CAVIAR context definition.

P_2: user defined variable. Determines a minimum value for those contexts that are not contained at all in any of the for the *H* hypotheses.

2.2 Weight Evaluation

The weight tag in the current XML Schema is related to the interestingness of the behaviour of an object. The interest ness for an object is based on:

1: The number of frames in the recent past that a person appears in the scene.

2: The interestingness of the movement in the current frame (user defined).

3: The interestingness of the context in the past frame (user defined).

According to the definition of these factors, a person that has been previously observed in the scene is more interesting than a person that just appeared. The interestingness of a movement is related to their activity level: persons moving exhibit more interesting behaviours than persons that remain immobile on the scene. The current descending order interestingness list is:

The interestingness of the contexts is related to security and commercial issues. Fighting contexts and contexts related to shopping activities are more interesting than persons remaining immobile in the scene. The current descending order interestingness list is:

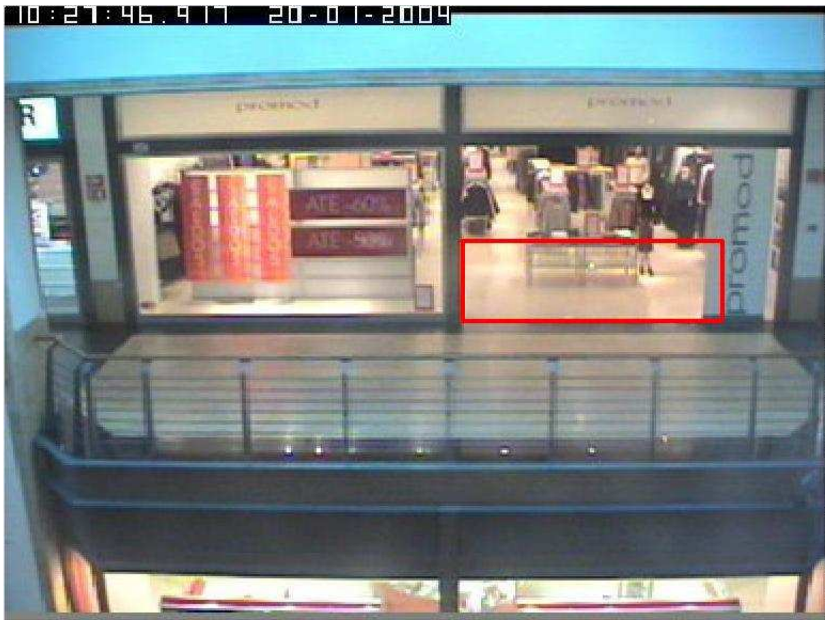

Figure 4: Entrance to Shop Areas for Portugal shopping centre (corridor view)

Person	Group
Running	Movement
Walking	Active
Active	Inactive
Inactive	

The interestingness value for contexts and movement is based on their position on the list as follows:

$$interest_x = 1 + scale_x * (N_x - pos_x) \quad x \in \{movement, context\}$$

Where N_x is the total of elements in *x* , *pos*is the position of *x* in the list, and *scale* is a user defined variable. The weight value for an object is computed by first computing the product of the factors:

$$weight\_pos = No\_Scenes * interest_{movement} * interest_{context}$$

Where *No_Scenes*is the number of scenes that the object has appeared in the recent past. Once *Weight_pos* is calculated for each object in the frame the objects are sorted out in descending order. Since *Weight_pos* is a value that is not normalized it's not sensible to use it as the weight for the object; instead the position of the object in the descending list (based on *Weight_pos*) determines the final weight for an object, as follows:

$$weight = 1 + scale_{weight} * (N_{objects} - pos)$$

Person	Group
drop down	fighting
browsing	meeting
immobile	browsing
window shop	window shop
shop exit	shop exit
shop enter	shop enter
shop reenter	shop reenter
walking	walking

Where:
$scale_{weight}$: User defined variable to determine the relative importance of a person.
$N_{objects}$: Total number of objects in the current frame.
pos : Position in the descending order list according to the value *Weight_pos*.
The probability of the contexts for an object and their weight are determined in separate processes. The value that determines the position for a pair (object,hypothesis) in the priority list is the product of the hypothesis probability and the weight of the object. The priority list is a descending order list.

2.3 Results

The goal of the filter is to minimize the time spent doing further analysis on the subsequent modules on false hypotheses. The filter produces a list where the most likely pairs (object, hypothesis) will be first analysed.

The evaluation of the module is based on an **evaluation score**. This is defined as the number of incorrect hypotheses analysed before the last correct hypothesis is analysed divided by the total number of objects in the current scene; ideally the evaluation value its zero. We also define a **local evaluation score** as the main position of each hypothesis amongst all hypotheses for each target. For example for the frame:
1 (object ID =1, hypothesis= walking) correct hypothesis
2 (object ID =1, hypothesis= immobile) false hypothesis
3 (object ID = 2, hypothesis= shop enter) false hypothesis
4 (object ID = 2, hypothesis= shop exit) true hypothesis
The evaluation for the frame its Eval=2/2=1.

The filtering module has only been tested for the ground truth labelled data. The filter has some variables defined as user defined. These variables have been tuned using a Monte Carlo Method. 8 sequences were used to train the algorithm and 4 were used for testing purposes. The algorithm was running using 40 elements per generation. The total of generations was 100. The initial value for each parameter and their ranges are illustrated in Table 1.

The new generation elements are created from the previous generation elements. Elements with best scores (smaller values) are more likely to be selected (according to a Bayesian distri-

Parameter	Initial Value	Range
Past Frames Analysed	5	4
P_1 (Hypotheses inference from past frame Hypotheses)	0.6	0.4
W_{zone} (Hypotheses inference from the position on the scene)	2.5	1.0
P_2 (Hypotheses inference from roles and movement on previous frames)	0.5	0.4
$Scale_{context}$	0.4	0.5
$Scale_{movement}$	0.4	0.5
$Scale_{weight}$	0.4	0.5

Table 1: Initial values for the Markovian algorithm

bution based on the scores). The best score obtained in the test was 0.1942 (local score 0.088), and the parameters for the evaluation are illustrated in Table 2.

Past Frames Analysed	5
P_1 (Hypotheses inference from past frame Hypotheses)	0.2
W_zone (Hypotheses inference from the position on the scene)	2.0
P_2 (Hypotheses inference from roles and movement on previous frames)	0.4
$Scale_context$	0.2
$Scale_movement$	0.2
$Scale_weight$	0.1

Table 2: Parameters for the best score

The worst score on the tests was 2.42 (local score 1.34) and the worst possible score its 7 (worst possible local score 7).

Although the value is very close to zero the tests were conducted only using ground truth labelled data for the previous context module (due to unavailability of the probabilistic module data). It is expected that once probabilistic information is used the scores will drop off a bit.

In order to assess the effect of the previous context module using the ground truth labelled data this module was removed from the Bayesian network and the scores obtained were 2.12 (local score 1.22). This verifies that the previous context has a strong effect in the performance of the filter; we expect that once the probabilistic module data is available the score for the tuned parameters will be in range 0.5-1.5.

3 Target Detection Prioritization

3.1 The components of the tracking system

The tracking system is composed of a central supervisor that calls subsequently the video demon, the robust tracking module and the target detection module (Figure 5). The supervisor manages

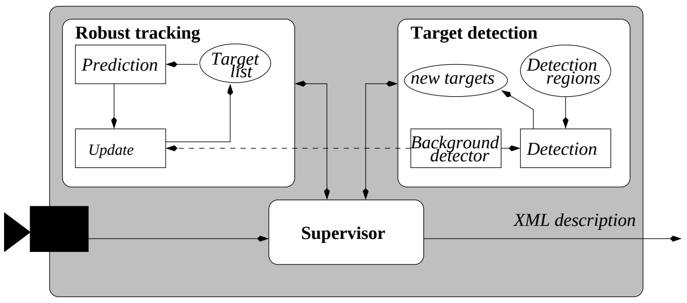

Figure 5: Architecture of the tracking and detection system controlled by a supervisor.

the data flow between the modules. The detection module detects new targets that are added to the target list. The tracking module provides robust tracking of the current targets using a Kalman filter. The supervisor stores the bounding boxes and orientation of the targets of each frame in a XML file using the CVML format [3].

The detection module uses a simple adaptive background differencing algorithm. We compared a number of background differencing algorithms in the joint article [2]. According to this study, the simple adaptive background differencing provides good results when it is combined with the tracking module. The current implementation of the target detection module uses a detection zones. These are rectangular regions within the image. The idea of using detection zones is to process only those regions where targets may appear. Typically, the surface of those regions is only a fraction of the surface of the image, which allows the detection to run faster.

We make the assumption that the scene is empty at the beginning of the sequence, and that new targets can enter only at predefined entry zones. The detection regions are placed such that they correspond to the physical entry zones in the scene. Under the condition that no targets appear elsewhere, this algorithm detects all targets reliably, detects less false targets, and can be computed much faster.

Naturally, the choice of the detection zones must be performed carefully, because the result of the detector depends on the choice of the detection zones. If targets appear at positions without a detection zone, they will not be detected. If a detection zone is placed within a zone of frequent lighting changes, those changes will produce false targets. In the default application the detection zones are determined manually. In the experiments, we compare several detection zone selection strategies with respect to computation time, detection rate and false alarm rate. The experiments give insight in the sensitivity of the tracking system to different strategies for the selection of detection zones.

3.2 Strategies for the selection of detection zones

There are several strategies to select detection zones for the detection module of the tracking system. The first strategy *(All)* consists in selecting all manually defined zones and process them at every frame. This is the default behaviour of the system and will serve as benchmark in the experiments (referred to as *5/1, (5 zones per 1 frame)* or *full*)

The second strategy *(Partial)* consists in selecting a subset of the manually defined detection zones at each frame. This subset can be selected at random or in a predefined order. This strategy should increase the processing speed and decrease slightly the number of detected targets, because some targets are detected a few frames later.

The third strategy *(Random)* consists in selecting at random a small subzone within the manually defined detection zones. The size of the subzone is chosen such that a target can be covered entirely. In the experiments we vary the number of selected subzones per frame.

The above strategies require the manual definition of the initial detection zones. To reduce the amount of manually defined data, we propose the following strategies that are fully automatic and require no a priori knowledge.

The fully automatic detection zone selection strategy *(Automatic)* selects random detection zones within the image. The size is fixed to 21 pixels squared. If the number of detection zones is sufficiently high, the probability of detecting all targets within the image is high. The experiments demonstrate this on an example. Naturally, successful detection requires a higher number of detection zones as the strategy *Random*, that uses the a priori knowledge of where targets may appear.

This fully automatic method can be used to learn automatically the entry zones of a new scene. In the last experiment *(Learned)*, the fully automatic method is combined with a feedback loop, that stores those random detection zones, that gave rise to a target. Ideally, these zones should coincide with the manually defined entry zones. The *Learned* selection strategy uses the learned detection zones and selects a subset at random from this list.

3.3 Experiment

The goal of this experiment is to study the effect of different selection strategies for the detection zones of the INRIA tracking system.

We evaluate the performance of the different selection strategies on the sequence *"Meet_Split_3rdguy.mpeg"* of the CAVIAR entry hall scenario. We chose this sequence, because of its relatively high complexity and the fact that 3 people enter at different entry zones. We are aware that testing this approach on only one sequence is too little experimental validation. Nevertheless, this gives a first insight on how the selection strategies perform.

The experiments are evaluated by comparing the annotated ground truth and the output of the XML writer of the tracking system. We compute for each frame the k pairs (true box, observed box) with highest overlap. We keep the pairs that have an overlap of more than 50%. The overlap is computed by

$$O(A_{true}, A_{obs}) = \frac{A_{true} \cap A_{obs}}{A_{true} \cup A_{obs}} \tag{1}$$

A pair is counted as correct (true positive) if $O(A_{true}, A_{obs}) < T = 50\%$. Any unmatched box of the ground truth is counted as a missed (false negative). Any unmatched box of the observation is counted as an insertion (false positive). From these values we compute the *Tracker detection rate (TRDR)* and the *False alarm rate (FAR)*. These are measures that were previously defined

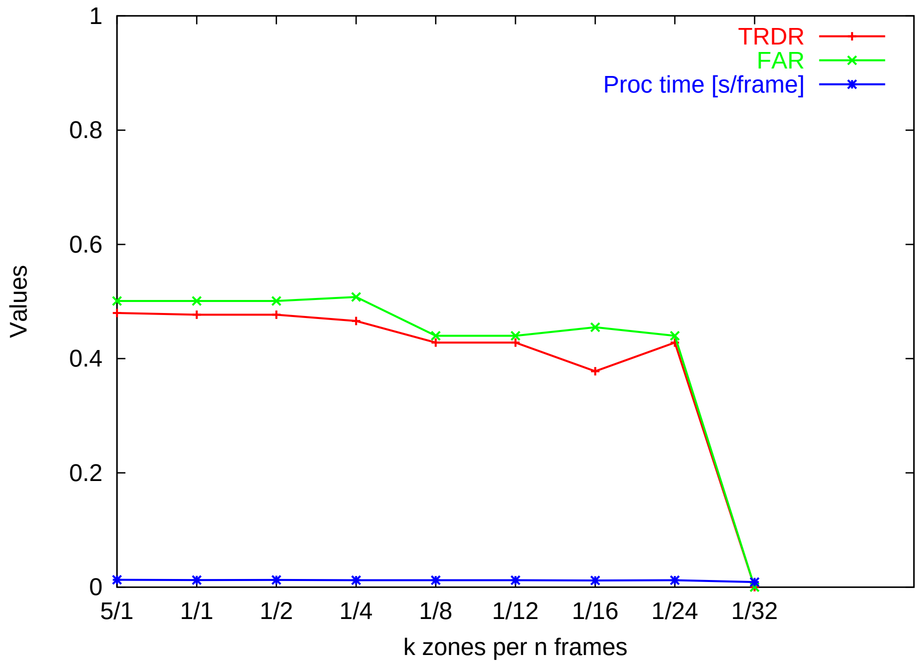

Figure 6: Performance of the detection and tracking system when only a subset of detection zones is processed.

by Black, Ellis and Rosin in [1] for the evaluation of tracking system performance.

$$TRDR = \frac{truepos}{truepos + falseneg}, FAR = \frac{falsepos}{truepos + falsepos} \tag{2}$$

The first experiment applies the strategy *Partial* where the manually defined detection zones are selected in a predefined order. The benchmark of the experiment is given by the run *(5/1)* that processes all 5 manually defined detection zones at every frame. We then reduce the number of processed detection zones per frame. We tested 1 zone per frame, 1 zone every two frames, to 1 zone every 32 frame. Figure 6 and Table 3 show the results.

The TRDR decreases with the number of detection zones. We have very good TRDR up to testing 1 zone every 4 frames. This means that every entry zone is processed once every 20 frames. The FAR increases little when the number of tested zones is reduced and decreases when we test less than 1 zone every 8 frames. The overall processing frequency increases from 77.8 Hz of the benchmark to 82.0 Hz of 1 zone every 4 frames to 82.9 Hz at 1 zone every 24 frames.

Targets move at a predefined maximum speed. In the sequence, this speed is rather small (about 200 pixels/s $\approx$ 6.7 pixels/frame). At this speed, a target takes an average of 4.5 frames to traverse a detection zone with width of 30 pixels. This explains the fact, that we catch all targets when testing one detection zone every 4 frames.

The second experiment evaluates the strategy *Random*. We generate a large list of overlapping subzones within the manually defined detection zones. The width is set to 21 pixels, and

Method	Frequency [Hz]	TRDR [%]	FAR [%]
5/1	77.8	48.0	50.1
1/1	81.1	47.7	50.1
1/2	78.6	47.7	50.1
1/4	82.0	46.6	50.8
1/8	81.7	42.8	44.0
1/12	82.8	42.8	44.0
1/16	86.2	37.8	45.5
1/24	82.9	42.8	44.0

Table 3: Performance of the detection and tracking system when only a subset of detection zones is processed.

Method	Frequency [Hz]	TRDR [%]	FAR [%]
full	77.8	48.0	50.1
20/1	48.5	47.8	43.4
15/1	50.2	47.9	43.4
10/1	52.0	47.7	43.5
5/1	53.1	47.5	43.5
4/1	54.2	47.6	43.5
3/1	53.7	47.5	43.6
2/1	54.3	47.4	43.7
1/1	54.9	44.2	44.8

Table 4: Performance of the detection and tracking system with randomly selected subzones within the detection zones.

the height equals the height of the original detection zone (between 20 and 30 pixels depending on the zone). The centers between two neighboring subzones is two pixels. From the 5 detection zones we obtain 116 subzones. The experiment draws randomly k subzones from this list and processes them. The TRDR and the FAR is displayed in Figure 7 and in Table 4. The results are the mean values of 5 runs. We observe little variance between the runs (example 5 zones per frame: $truepos \in [822, 827], \sigma(truepos) = 1.6, falsepos \in [635, 637], \sigma(falsepos) = 0.6$).

The TRDR stays constant up to testing only 2 subzones per frame. It drops from 48.0% to 44.2% when testing only 1 subzone per frame. The surprising result is that the random method has a smaller FAR. This may be interpreted that the subzones are less sensitive to false detections. This result needs to be confirmed with a larger experiment. The function that allows to draw at random from a list of elements seems to generate an overhead. Whereas we process a smaller total surface of detection zones, the computation time is higher than in the previous experiment. We have implemented the experiment straight forward. Optimization may cause a speed up.

The third experiment evaluates the performance of detection and tracking system when detection zones are chosen randomly all over the image. The detection zones have a fixed size 21

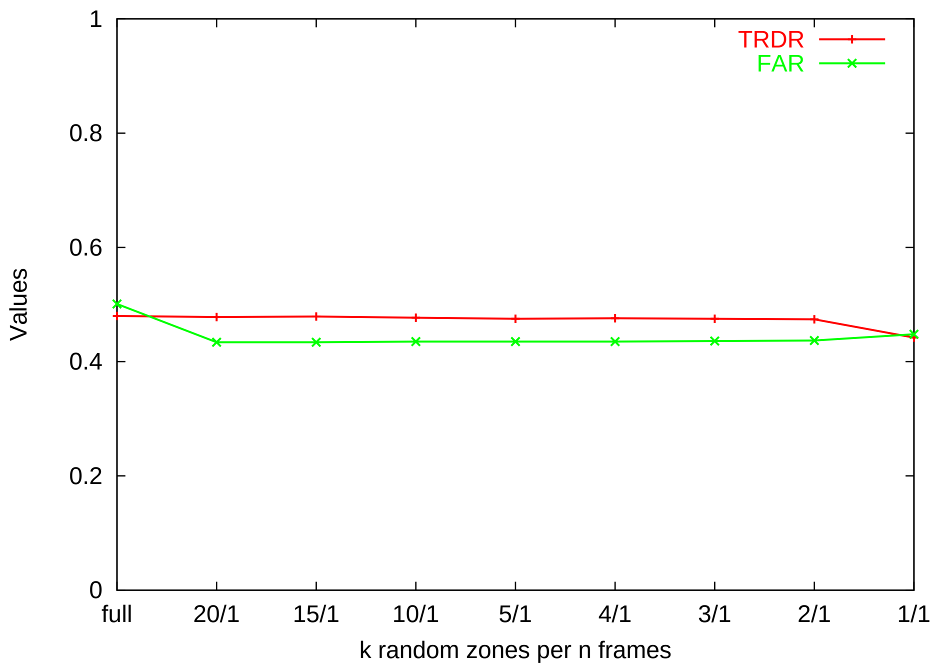

Figure 7: Performance of the detection and tracking system with randomly selected subzones within the detection zones.

pixels squared. We generate a list of equally space detection zones with distance of 5 pixels all over the image. This gives a total of $55 \times 74 = 4070$ zones. We draw at random k detection zones from this list and process them. The results are shown in Figure 8 and Table 5. The values are averaged over 3 runs. The TRDR is significantly inferior to the benchmark. We observe also a much higher FAR. The best result is obtained using 10 zones per frame (TRDR=33.6% and FAR=72.2%). Using a higher number of zones may increase the TRDR, but would also increase the FAR. A reason for the high FAR may be that there are persons in the scene at the beginning of the sequence. These persons are nearly immobile, and are therefore not labelled in the ground truth. The small movements of those persons may cause some of the false positives. The processing time is not reduced. There is a certain overhead for the random selection from the list of subzones. This method has still the advantage, that it does not require the manual definition of the detection zones. This method is fully automatic.

The fourth experiment shows how the *Automatic* strategy can be used for learning the entry zones. The experiment is then performed by drawing at random from the learned entry zones. We use 3 sequences for learning the entry zones by applying the *Automatic* strategy. Every time a detection zone gives rise to a target, this zone is stored. From the 3 sequences “Browse1”, “Meet_Crowd” and “Meet_WalkTogether1” 12 detection zones are obtained. They cover the three most important entry zones of the scene. Figure 9 and Table 6 show the result of drawing at random from this list.

The TRDR is still inferior to the benchmark, but the FAR is smaller than in the experiment

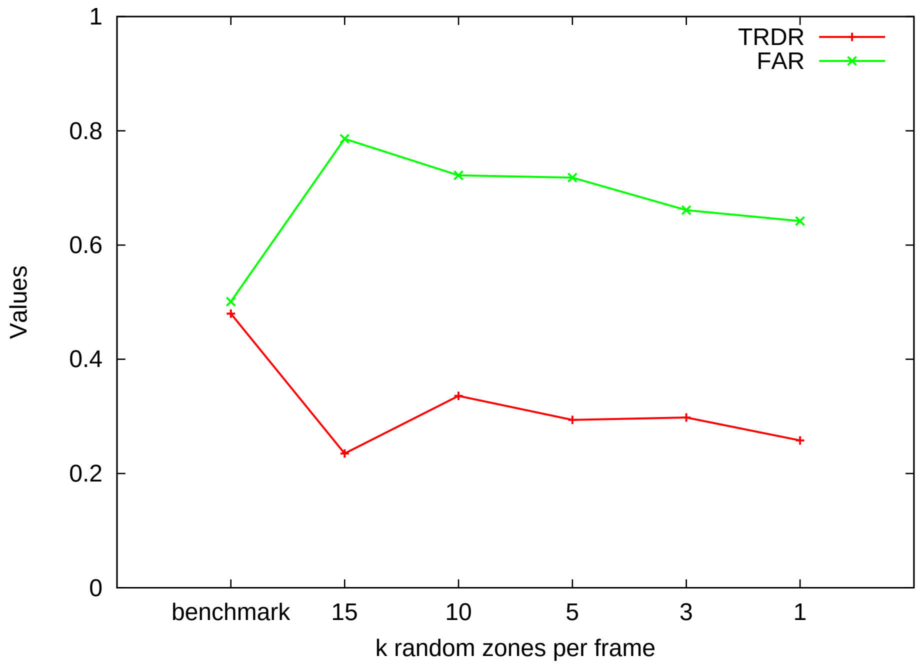

Figure 8: Performance of the detection and tracking system with randomly selected subzones within the entire image.

Method	Frequency [Hz]	TRDR [%]	FAR [%]
full	77.8	48.0	50.1
15/1	17.7	23.5	78.6
10/1	22.5	33.6	72.2
5/1	30.2	29.4	71.8
3/1	36.7	29.8	66.1
1/1	46.5	25.8	64.2

Table 5: Performance of the detection and tracking system with randomly selected subzones within the entire image.

Method	Frequency [Hz]	TRDR [%]	FAR [%]
full	77.8	48.0	50.1
12/1	46.7	33.5	67.4
10/1	46.8	34.0	66.9
5/1	48.8	33.6	67.3
1/1	49.9	30.5	67.8

Table 6: Performance of the detection and tracking system with randomly selected subzones among the entry zones learned from 3 training sequences by the automatic approach.

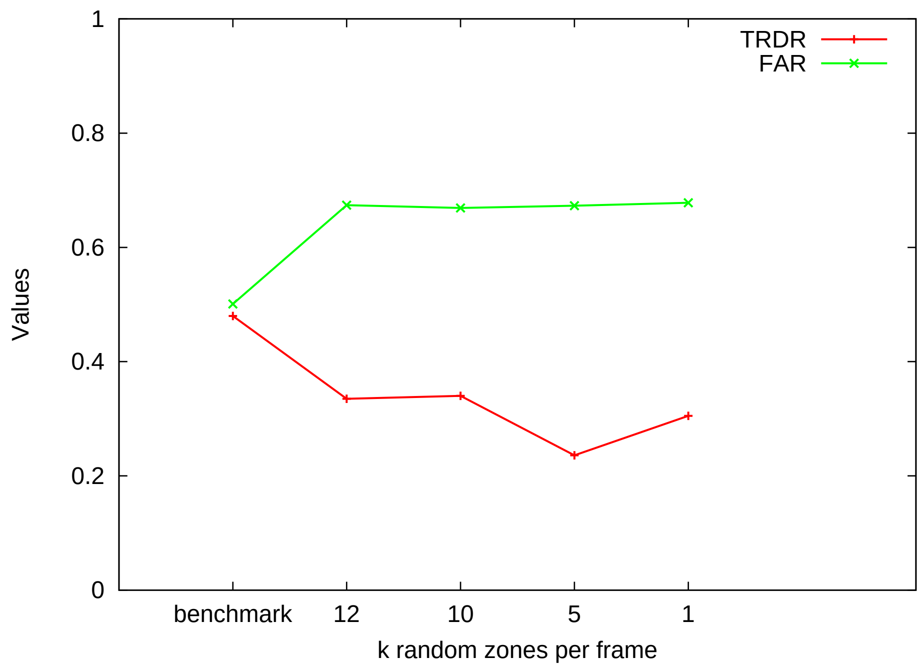

Figure 9: Performance of the detection and tracking system with randomly selected subzones among the entry zones learned from 3 training sequences by the automatic approach.

Automatic. A reason may be found that the detection zones are not as dense as in the strategy *Random*. A longer learning phase would produce a higher number of detection zones, that may be more dense and would increase the results. The nice point about this method is that it can be generated automatically without human supervision.

3.4 Conclusion

We showed that the detection module of the detection and tracking system depends strongly on the choice of the detection zones. We defined several strategies on how to select the zones and evaluated the results with respect to annotated ground truth. The best result is obtained by the strategy *Random* where small subzones within manually defined entry zones are drawn at random. We obtain an equal TRDR as the benchmark and the FAR decreases. A possible explanation is that subzones are less sensitive to false detections due to their size.

We also experimented with two fully automatic selection strategies. The performance is significantly below the benchmark. More experiments should clarify, if a longer learning phase would increase the performance.

References

[1] J. Black, T. Ellis, and P. Rosin. A novel method for video tracking performance evaluation. In *International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 125–132, 2003.

[2] D.Hall, J. Nascimento, P. Ribeiro, E. Andrade, P. Moreno, S. Pesnel, T. List, R. Emonet, R.B. Fisher, J. Santos Victor, and J.L. Crowley. Comparison of target detection algorithms using adaptive background models. In *International Workshop on Performance Evaluation of Tracking and Surveillance*, 2005. to appear.

[3] T. List and R.B. Fisher. CVML- an XML-based computer vision markup language. In *International Conference on Pattern Recognition*, pages I 789–792, Cambridge, UK, August 2004.