Attention Manipulation for Naval Tactical Picture Compilation

Tibor Bosse[1], Rianne van Lambalgen[1], Peter-Paul van Maanen[1,2], and Jan Treur[1]

[1] *Vrije Universiteit Amsterdam, Amsterdam, The Netherlands*
{tbosse,rm.van.lambalgen,treur}@cs.vu.nl
[2] *TNO Human Factors, Soesterberg, The Netherlands*
peter-paul.vanmaanen@tno.nl

Abstract

This paper presents an agent model that is able to manipulate the visual attention of a human, in order to support naval crew. It consists of four submodels, including a model to reason about a subject's attention. A practical case study was formally analysed, and the results were verified using automated checking tools. Results show how a human subject's attention is manipulated by adjusting luminance, based on assessment of the subject's attention. A first evaluation of the agent shows a positive effect.

1. Introduction

In the domain of naval warfare, it is crucial for the crew of the vessels involved to be aware of the situation in the field. One of the crew members is usually assigned the task to identify and classify all entities in the environment (e.g., [6]). This task determines the type and intent of a multiplicity of contacts on a radar screen. Attention is typically directed to one bit of information at a time [13], [14], [16]. A supporting software agent may alert the human about a contact if it is ignored. To this end the agent has to maintain a model of the cognitive state of the human including the human's distribution of attention. Existing cognitive models on attention show that it is possible to predict a person's attention based on a saliency map, calculated from features of a stimulus, like luminance, colour and orientation [7], [12]. In this study, a Theory of Mind (or ToM, e.g., [4]) model is exploited within the agent model to analyse attention of the human. Attention can then be influenced (or 'manipulated') by changing features of stimuli, e.g., its contrast with stimuli at other locations [7], [8], [11], its luminance [15], [17], or its form [17].

Some approaches in the literature address the development of software agents with a Theory of Mind (e.g., [4], [9], [10]), but only address a model of the epistemic (e.g., beliefs), motivational (e.g., desires, intentions), and/or emotional states of other agents. For the situation sketched above, attribution of attentional states has to be addressed. In the current paper, an agent model has been developed, which uses four specific (sub)models. The first is a representation of a dynamical model of human attention, for estimation of the locations of a person's attention, based on information about features of objects on the screen and the person's gaze. The second model is a reasoning model which the agent uses to reason through the first model, to generate beliefs on attentional states at any point in time. With a third model the agent compares the output of the second model with a normative attention distribution and determines the discrepancy. Finally, a fourth model is used to direct the person's attention to relevant contacts based on the output of the third model.

Initial versions of the first two models were adopted from earlier work [5]. The current paper focuses on the last two models. Section 2 shortly describes a formalisation of the different models. In Section 3 a case study is shown where the software agent is used to manipulate a subject's attention. Based on this case study, Section 4 addresses experimental validation of the results, and Section 5 automated verification of different important properties of the submodels used in the agent. In Section 6, a formal mathematical analysis of the model is given. Finally, Section 7 is a discussion.

2. A Theory of Mind for Attention

This section introduces the ambient agent with a Theory of Mind on attention and the four different models which are used within this agent: Dynamic Attention Model, Model for Beliefs about Attention, Model to Determine Discrepancy and Decision Model for Attention Adjustment. The agent and its interaction with the environment (involving a complex task and an eye-tracker, see Section 3) are schematically displayed in Figure 1.
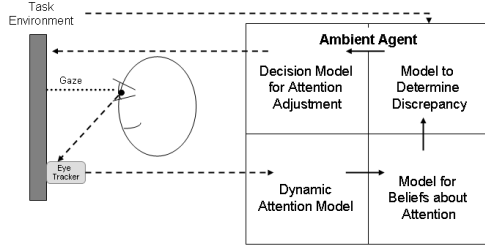
Figure 1. Overview of the ambient agent and its environment

2.1. The Dynamic Attention Model used

This model is taken over from [5] and is only briefly summarised in this section. The model uses three types of input: information about the human's *gaze direction*, about *locations* (or spaces) and about *features* of objects on the screen. Based on this, it makes an estimation of the *current attention distribution* at a time point: an assignment of attention values AV(s, t) to a set of attention spaces at that time. The attention distribution is assumed to have a certain persistency. At each point in time the new *attention level* is related to the previous attention, by

$$AV(s,t) = \lambda \cdot AV(s,t-1) + (1 - \lambda) \cdot AV_{norm}(s,t)$$

Here, λ is the decay parameter for the decay of the attention value of space s at time point $t - 1$, and $AV_{norm}(s,t)$ is determined by *normalisation* for the total amount of attention A(t), described by:

$$AV_{norm}(s,t) = \frac{AV_{new}(s,t)}{\sum_{s'} AV_{new}(s',t)} \cdot A(t)$$

$$AV_{new}(s,t) = \frac{AV_{pot}(s,t)}{1 + \alpha \cdot r(s,t)^2}$$

Here $AV_{new}(s, t)$ is calculated from the potential attention value of space s at time point t and the relative distance of each space s to the gaze point (the centre). The term r(s,t) is taken as the Euclidian distance between the current gaze point and s at time point t (multiplied by an importance factor α which determines the relative impact of the distance to the gaze point on the attentional state, which can be different per individual and situation):

$$r(s,t) = d_{eucl}(gaze(t),s)$$

The potential attention value $AV_{pot}(s,t)$ is a weighted sum of the features of the space (i.e., of the types of objects present) at that time (e.g., luminance, colour):

$$AV_{pot}(s,t) = \sum_{maps\ M} M(s,t) \cdot w_M(s,t)$$

For every feature there is a saliency map M, which describes its potency of drawing attention (e.g. [7], [9], [17]). Moreover, M(s,t) is the unweighted potential attention value of s at time point t, and $w_M(s,t)$ is the weight used for saliency map M, where $1 \leq M(s,t)$ and $0 \leq w_M(s,t) \leq 1$.

Figure 2 shows an overview of this model. The circles denote the italicised concepts introduced above, and the arrows indicate influences between concepts.

2.2. Model for Beliefs about Attention

This (reasoning) model is used to generate beliefs about attentional states of the other agent. The software agent uses the dynamical system model as described in Section 2.1 as an internal simulation model to generate new attentional states from the previous ones, gaze information and features of the object, with the use of a forward reasoning method (forward in time) as described in [2]. The basic specification of the reasoning model can be expressed by the representation leads_to_after(I, J, D) (belief that I leads to J after duration D). Here, I and J are both information elements (i.e., they may correspond to any concept from Figure 2, e.g., gaze_at(1, 2) or has_value(av(1,2), 0.68).

In addition, the representation at(I, T) gives information on the world (including human processes) at different points in time. It represents a belief that state I holds at time point T. For example, at(gaze_at(1,2), 53) expresses that at time point 53, the human's gaze is at the space with coordinates {1,2}.
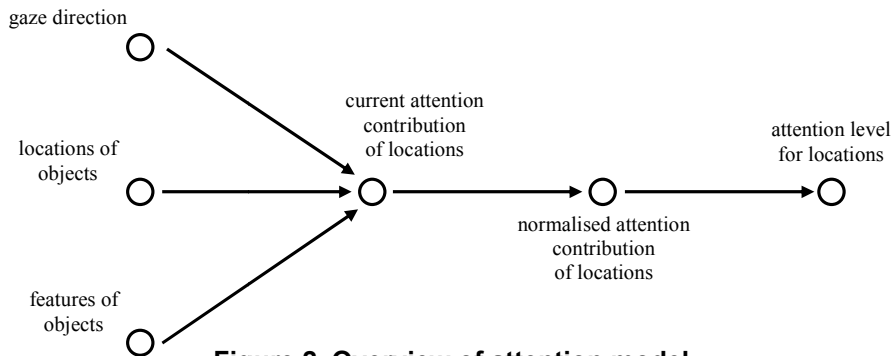
Figure 2. Overview of attention model.

2.3. Model to Determine Discrepancy

With this model the agent determines the discrepancy between actual and desirable attentional states and to what extent the attention distribution has to change. This is based on a model for the desirable attention distribution (prescriptive model). For the case addressed this means an assessment of which objects deserve attention (based on features as distance, speed and direction). To be able to make such assessments, the agent is provided with some tactical domain knowledge, in terms of heuristics (also see Section 3.1)

2.4. Decision Model for Attention Adjustment

Given a desire to adjust the attention distribution, the agent uses this model to determine how the inputs for the attention model have to be changed. Besides output from the model in Section 2.3, it uses the structure of the model described in Section 2.1. Relations between variables within the latter model are followed in a backward manner, thereby propagating the desired adjustment from the attentional state variable to the features of the object at the screen. This is a form of desire refinement: starting from a (root) variable, step-by-step a desire on adjusting a (parent) variable is refined to desires on adjustments of the (children) variables on which the (parent) variable depends, until the leave variables are reached. Starting point is the adjustment of the attentional state $av(s)$, see the following reasoning rule:

desire(av(s)>h) $\land$ belief(has_value(av(s), v)) $\land$ v<h $\rightarrow$ desire(adjust_by(av(s), (h-v)/v))

Note that here the adjustment is taken relative. Suppose as a point of departure an adjustment $\Delta v1$ is desired, and that $v1$ depends on two variables $v11$ and $v12$ that are adjustable (the non-adjustable variables can be left out of consideration). Then by elementary calculus the following linear approximation can be obtained:

$$\Delta v_1 = \Delta v_{11} + \Delta v_{12}$$

This is used to determine the desired adjustments $\Delta v11$ and $\Delta v12$ from $\Delta v1$, where by weight factors $\mu11$ and $\mu12$ the proportion can be indicated in which the variables should contribute to the adjustment: $\Delta v11/\Delta v12 = \mu11/\mu12$. Since then

$$\Delta v_1 = \frac{\partial v1}{\partial v_{11}} \Delta v_{12}\, \mu_{11}/\mu_{12}\ + \frac{\partial v1}{\partial v_{12}}\ \Delta v_{12} = (\frac{\partial v1}{\partial v_{11}}\, \mu_{11}/\mu_{12}\ +\ \frac{\partial v1}{\partial v_{12}}\ )\, \Delta v_{12}$$

the adjustments can be made as follows:

$$\Delta v_{12} = \frac{\Delta v1}{\frac{\partial v1}{\partial v_{11}} \mu11/\mu12\ +\ \frac{\partial v1}{\partial v_{12}}}$$

$$\Delta v_{11} = \frac{\Delta v1}{\frac{\partial v1}{\partial v_{11}}\ +\ \frac{\partial v1}{\partial v_{12}} \mu12/\mu11}$$

Special cases are $\mu11 = \mu12 = 1$ (absolute equal contribution) or $\mu11 = v11$ and $\mu12 = v12$ (relative equal contribution: in proportion with their absolute values). As an example, consider a variable that is the weighted sum of two other variables (as is the case, for example, for the aggregation of the effects of the features of the objects on the attentional state): $v1 = w11v11 + w12v12$. For this case, the partial derivatives are $w11$ respectively $w12$; therefore

$$\Delta v_{11} = \frac{\Delta v1}{w11\ +\ w12\, \mu12/\mu11} \qquad \Delta v_{12} = \frac{\Delta v1}{w11\, \mu11/\mu12\ +\ w12}$$

When $\mu11 = \mu12 = 1$ this results in $\Delta v11 = \Delta v12 = \Delta v1/(w11 + w12)$, and when in addition $w11 + w12 = 1$, it holds $\Delta v11 = \Delta v12 = \Delta v1$. Another setting, which actually has been used in the model, is to take $\mu11 = v11$ and $\mu12 = v12$. In this case the adjustments are assigned proportionally; for example, when $v1$ has to be adjusted by 5%, also the other two variables on which it depends need to contribute an adjustment of 5%. Thus the relative adjustment remains the same through propagations:

$$\frac{\Delta v11}{v11} = \frac{\Delta v1}{w11\ +\ w12\, v12/v11}/v_{11} = \frac{\Delta v1}{v1}$$

This shows the general approach on how desired adjustments can be propagated in a backward manner. For the case study undertaken, a desired adjustment of the attentional state is related to adjustments in the features of the displayed objects. An example of a (temporal logical) reasoning rule specified to achieve this propagation process is:

desire(adjust_by(u1, a)) $\land$ belief(depends_on(u1, u2)) $\rightarrow$ desire(adjust_by(u2, a))

Here the adjustments are taken relative, so, this rule is based on $\Delta u2 / u2 = \Delta u1 / u1$ as derived above. When at the end the leaves are reached, (represented by the belief that they are directly adjustable), an intention to adjust them is derived.

desire(adjust_by(u, a)) $\land$ belief(directly_adjustable(u)) $\rightarrow$ intention(adjust_by(u, a))

If an intention to adjust a variable u by a exists with current value b, the new value $b + \alpha*a*b$ to be assigned to u is determined; here α is a parameter that allows tuning the speed of adjustment:

intention(adjust_by(u, a)) $\land$ belief(has_value_for(u, b)) $\rightarrow$ performed(assign_new_value_for(u, b+ αa*b))

This rule is applied for variables that describe features f of objects at spaces s (instances for u of the form feature(s, f)). Note that the adjustment is propagated as a value relative to the overall value.

3. Case Study

To test the approach in a real-world situation, a case study with human subjects while executing the Tactical Picture Compilation Task has been undertaken. In Section 3.1 the environment is explained, Section 3.2 discusses some implementation details of the agent tailored to the environment, and in Section 3.3 the results are described based on different examples.
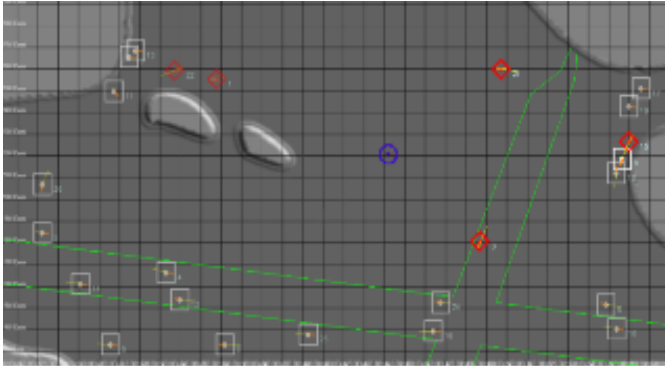
Figure 3. Interface of the task environment

3.1. Environment

The task used for this case study is an altered version of the identification task described in [11] that has to be executed in order to build up a tactical picture of the situation, i.e. the Tactical Picture Compilation Task (TPCT). The implementation of the software was done in Gamemaker [19]. In Figure 3 a snapshot of the interface of the task environment is shown. The goal is to identify the five most threatening contacts (ships). In order to do this, participants monitor a radar display of contacts in the surrounding areas. To determine if a contact is a possible threat, different criteria have to be used. These criteria are the identification criteria (idcrits) that are also used in naval warfare, but are simplified in order to let naive participants learn them more easily. These simplified criteria are the speed (depicted by the length of the tail of a contact), direction (pointer in front of a contact), distance of a contact to the own ship (circular object), and whether the contact is in a sea lane or not (in or out the large open cross). Contacts can be identified as either a threat (diamond) or no threat (square).

3.2. Implementation

The agent was further developed and evaluated using Matlab. The output of the environment described in Section 3.1 was used and consisted of a representation of all properties of the contacts visible on the screen, i.e. speed, direction, if it is in a sea lane or not, distance to the own ship, location on the screen and contact number. In addition, data from a Tobii x50 eye-tracker [18] were retrieved from a participant executing the TPC task. All data were retrieved several times per second and were used as input for the models within the agent. Once the model for adjustment of attention (model 4) was tailored to the TPC case study, the eventual implementation of it was done in C#. The output of the implementation causes the saliency of the different objects on the screen to either increase or decrease, which may result in a shift of the participant's visual attention. As a result, the participant's attention is continuously manipulated in such a way that it is expected that he pays attention to the objects that are considered relevant by the agent. The increase or decrease of the saliency of objects can be done on a continuous or discrete scale, with a binary scale as being discrete. An illustration of the output of this implementation with attention manipulation on a continuous scale is described below. The results of the implementation with attention manipulation on a binary scale have been omitted for reasons of space, but are investigated nonetheless in Sections 5 and 6.

3.3. Results

The first results of the agent implemented for this case study are best described by a number of example snapshots of the outcomes of the models used in the agent to estimate (model 1) and manipulate (model 4) attention in three different situations over time (see Figure 4).

On the left side of Figure 4 the darker dots correspond to the agent's estimation of those contacts to which the participant is paying attention. On the right side of the figure, the darker dots correspond to those contacts where attention manipulation is initiated by the system (in this case, by increasing its saliency). On both sides of the figure a cross corresponds to the own ship, a star corresponds to the eye point of gaze, and the x- and y-axes represent the coordinates on the interface of the TPCT. In the pictures to the left, the z-axis represents the estimated amount of attention. The darker dots on the left side are a result of the exceedance of this estimation of a certain threshold (in this case .03). Thus, a peak indicates that it is estimated that the participant has attention for that location. Furthermore, from top to bottom, the following three situations are displayed in Figure 4:

1. After 37 seconds since the beginning of the experiment, the participant is not paying attention to region A at coordinates (7.5,1.5) and no

attention manipulation for region A is initiated by the system.

2. After 39 seconds, the participant is not paying attention to region A, while the attention should be allocated to region A, and therefore attention manipulation for region A is initiated by the system.

3. After 43 seconds, the participant is paying attention to region A, while no attention manipulation for region A is done by the system, because this is not needed anymore.
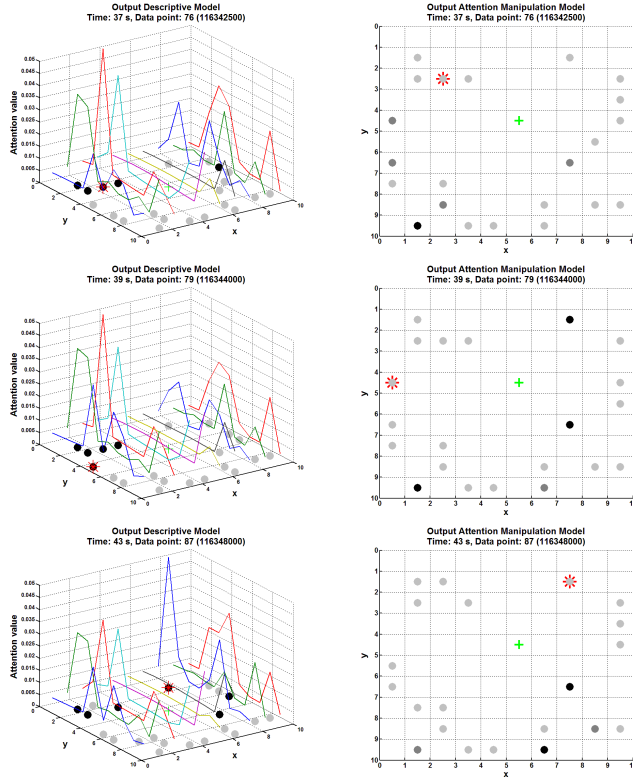
Figure 4. Estimation of the participant's attention division and the agent's reaction

The output of the attention manipulation system and the resulting reaction in terms of the allocation of the participant's attention in the above three situations, show what one would expect of an accurate system of attention manipulation. As shown in the two pictures at the bottom of Figure 4, in this case the agent indeed succeeds in attracting the attention of the participant: both the gaze (the star in the bottom right picture) and the estimated attention (the peak in the bottom left picture) shift towards the location that has been manipulated.

4. Validation

In order to validate the agent's manipulation model, the results from the case study have been used and tested against results that were obtained in a similar setting without manipulation of attention. The basic idea was to show that the agent's manipulation of attention indeed results in a significant improvement of human performance. Human performance in selecting the five most threatening contacts was compared during two periods of 10 minutes (with and without manipulation, respectively). The type of manipulation was based on determining the saliency of the objects on a binary scale. In this way it was easy (opposed to a continuous scale) to follow the agent's advice. The performance measure took the severity of an error into account. Taking the severity into account is important, because for instance selecting the least threatening contact as a threat is a more severe error than selecting the sixth most threatening contact. This was done by the use of the following penalty function (P):

$$P_x = \frac{p_x}{\sum_k^{24} p_k} = \frac{abs(t_x - \frac{t_5 + t_6}{2})}{\sum_k^{24} p_k}$$

where p_x is the prenormalised penalty of contact x and t_x is the threat value of contact x (there are 24 contacts). Human performance is then calculated by adding all penalties of the contacts that are incorrectly selected as one of the top five threats and subtracting them from 1.

After the above alterations, the average human performance over all time points of the condition "support" was compared with the average human performance of the first condition "no support", where "support" (M+ = .8714, SD+ = .0569) was found significantly higher than "no support" (M− = .8541, SD− = .0667), with t(df = 5632) = 10.46, p = 0. Hence significant improvements were found comparing the first and the second condition. Finally, subjective data based on a questionnaire pointed out that the participant preferred the "support" condition above that of the "no support" condition.

5. Verification

In addition to this validation, the results of the experiment have been analysed in more detail by converting them into formally specified *traces* (i.e., sequences of events over time), and checking relevant *properties*, expressed as temporal logical expressions, against these traces. To this end, a number of properties were logically formalised in the language TTL [3]. This predicate logical language supports formal specification and analysis of dynamic

properties. TTL is built on atoms referring to states of the world, time points and traces, i.e. trajectories of states over time. In addition, dynamic properties are temporal statements that can be formulated with respect to traces based on the state ontology Ont in the following manner. Given a trace γ over state ontology Ont, the state in γ at time point t is denoted by state(γ, t). These states can be related to state properties via the formally defined satisfaction relation denoted by the infix predicate |=, comparable to the Holds-predicate in the Situation Calculus: state(γ, t) |= p denotes that state property p holds in trace γ at time t.

Based on these statements, dynamic properties can be formulated in a formal manner in a sorted first-order predicate logic, using quantifiers over time and traces and the usual first-order logical connectives such as $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\forall$, $\exists$. To give a simple example, the property 'there is a time point t in trace 1 at which the estimated attention level of space {1,2} is 0.5' is formalised as follows (see [3] for more details):

$\exists$t:TIME   state(trace1,t) |= belief(has_value(av(1,2), 0.5))

Below, a number of such dynamic properties that are relevant to check the agent's attention manipulation are formalised in TTL. To this end, some abbreviations are defined:

discrepancy_at(γ:TRACE, t:TIME, x,y:COORDINATE) $\equiv$
  $\exists$a,h:REAL estimated_attention_at(γ,t,x,y,a) &
  state(γ,t) |= desire(has_value(av(x,y), h)) & a<h

This predicate states that at time point t in trace γ, there is a discrepancy at space {x,y}. This is the case when the estimated attention at this space is smaller than the desired attention. Next, abbreviation estimated_attention_at is defined:

estimated_attention_at(γ:TRACE,t:TIME,
  x,y:COORDINATE, a:REAL)
  $\equiv$ state(γ,t) |= belief(has_value(av(x,y),a))

This takes the estimated attention as calculated by the agent at runtime. This means that this definition can only be used under the assumption that this calculation is correct. Since this is not necessarily the case, a second option is to calculate the estimated attention during the checking process, based on more objective data such as the gaze data and the features of the contacts.

Based on these abbreviations, several relevant properties may be defined. An example of a relevant property is the following (note that this property assumes a given trace γ, a given time point t, and a given space {x,y}):

PP1 (Discrepancy leads to Efficient Gaze Movement)
If there is a discrepancy at {x,y} and the gaze is currently at {x2,y2}, then within δ time points the gaze will have moved to another space {x3,y3} that is closer to {x,y} (according to the Euclidean distance).
PP1(γ:TRACE, t:TIME, x,y:COORDINATE) $\equiv$
$\forall$x2,y2:COORDINATE
discrepancy_at(γ,t,x,y) &
state(γ,t) |= gaze_at(x2,y2) & t < LT-δ
  $\Rightarrow$ $\exists$t2:TIME $\exists$x3,y3:COORDINATE [t<t2<t+δ &
  state(γ,t2) |= gaze_at(x3,y3) &
  $\sqrt{((x-x2)^2+(y-y2)^2)} > \sqrt{((x-x3)^2+(y-y3)^2)}$]

In the above property, a reasonable value should be chosen for the delay parameter δ. Ideally, δ equals the sum of 1) the time it takes the agent to adapt the features of the contacts and 2) the person's reaction time.

To enable automated checks, a special software environment for TTL exists, featuring both a Property Editor for building and editing TTL properties and a Checking Tool that enables formal verification of such properties against traces [3]. Using this TTL Checking Tool, properties can be automatically checked against traces generated from any case study. In this paper the properties were checked against the traces from the experiment described in Section 5. When checking such properties, it is useful to know not only if a certain property holds for a specific space at a specific time point in a specific trace, but also how often it holds. This will provide a measure of the successfulness of the system. To check such more statistical properties, TTL offers the possibility to test a property for all time points, and sum the cases that it holds. Via this approach, PP1 was checked against the traces of the experiment with δ = 3.0 sec. These checks pointed out that (under the "support" condition) in 88.4% of the cases that there was a discrepancy, the gaze of the person changed towards the location of the discrepancy. Under the "no support" condition, this was around 80%.

6. Formal Analysis

The results of validation and verification discussed above may ask for a more detailed analysis. In particular, the question may arise of how a difference between 80% without support and 88% with support as reported above should be interpreted. Here a more detailed formal analysis is given that supports the context for interpretation of such percentages. To this end the effect of arbitrary transitions in gaze dynamics is analysed, in particular those that occur between the time points of monitoring the gaze and adjustment of luminance.

At a given time point, the adjustment of luminance is based on the gaze at that point in time. A question is whether at the time the luminance is actually adjusted,

the gaze is still at the same point. When the system is very fast in adjusting the luminance this may be the case. However, it is also possible that even in this very short time the gaze has changed to focus on another location on the screen. Here it is analysed in how many cases of an arbitrarily changed gaze still the luminance adjustment by the system should be sufficient. The general idea is that this is the case as long as the gaze transition does not increase the distance between gaze location and considered discrepancy location. The area of all locations of the screen for which this is the case is calculated mathematically below; here the worst case is analysed, the case when the considered discrepancy location is at the corner of the screen. The screen is taken as a square. The function f indicates an under-approximation of the number (measured by the area) of locations with distance at most r of O. For $r \leq d$ the area within distance r of O is a quarter of a circle: $\pi/4$ $r2$; so $f(r) = \pi/4\ r2$ for $r \leq d$. For $r>d$ an approximation was made. The part of distance larger to O than r is approximated by two triangles as ΔPQR in Figure 5.

$ON = d\ QN = \sqrt{(r^2 - d^2)}\ QR = RN - QN = d - \sqrt{(r^2 - d^2)}$
$OR = \sqrt{2}*d \qquad PR = OR - OP = \sqrt{2}*d - r$
$\Delta PQR = \frac{1}{2}\ PR * h$ with h the distance of Q to OR.
$h = \frac{1}{2}\ \sqrt{2}*QR = \frac{1}{2}\ \sqrt{2} * (d - \sqrt{(r^2 - d^2)})$
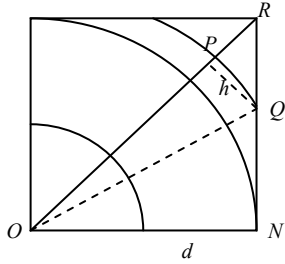
**Figure 5. Gaze area approximation**

The whole area $-2\Delta PQR$ is

$d^2 - \frac{1}{2}\ \sqrt{2} * (d - \sqrt{(r^2 - d^2)})(\sqrt{2}*d - r)$

Therefore it is taken

$f(r) = d^2 - \frac{1}{2}\ \sqrt{2} * (d - \sqrt{(r^2 - d^2)})(\sqrt{2}*d - r)$ for $r>d$

For $d = 10$ the overall function f divided by the overall area d2 (thus normalising it between 0 and 1) is shown in Figure 6. For example, it shows that when $r = \frac{1}{2}d$, then the covered area is around 20% of the overall screen, but when r is a bit larger, for example $r = d$, then at least around 80% is covered. Note that this is a worst case analysis with the location considered in the corner. In less extreme cases the situation can differ. When, for example, the considered location is at the center, then for distance $r = \frac{1}{2}\ d$, the covered area

would be a full circle with radius ½d, so an area of $\pi/4$ d2, which is more than 70% of the overall area.
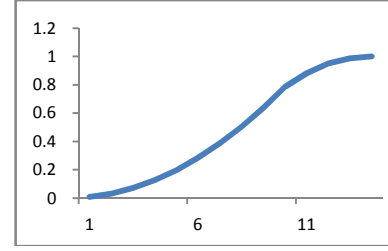
Figure 6. Function of the number of locations within distance r of *O*, divided by *d*^{2}, for d=10

Moreover, the distance of the considered location where a discrepancy is detected to the actual gaze may not have a uniform probability distribution from 0 to $\sqrt{2}*d$. Indeed, the value 0 may be very improbable, and the larger values may have much higher probabilities. Suppose p(r) denotes the probability (density) that the distance between actual gaze and considered discrepancy location is r, then the expected coverage can be calculated by:

$$\int_0^{\sqrt{2}*d} p(r) * f(r) dr$$

For example, if a probability distribution is assumed that is increasing in a modest, linear way from p(0) = 0 to p($\sqrt{2}*d$) = 1/d2, then for d = 10 with p(r) = r/100 this becomes approximately (estimated by numerical integration):

$$\int_0^{14} \frac{r*f(r)}{100} dr = 0.72$$

This means that the expected coverage would be 72%. For a bit less modest increase, for example in a quadratic manner for d = 10 from p(0) = 0 to p(14) = 0.2, then the expected coverage is approximately 80% (estimated by numerical integration):

$$\int_0^{14} \frac{r2*f(r)}{1000} dr = 0.80$$

When it turns out that the gaze is often changing, then a remedy is to base the adjustment of the luminance on a larger distance for r, thus anticipating on the possible future states. The graph for f shows that if r is taken equal to distance d, then a coverage of 80% is achieved.

7. Discussion

An important task in the domain of naval warfare is the Tactical Picture Compilation Task, where persons have to deal with a lot of complex and dynamic information at the same time. To obtain an optimal performance, an intelligent agent can provide aid in such a task. This paper presented an initial version of

such a supporting software agent. Within this type of agent an explicitly represented model of human functioning plays an important role, for the case considered here the model of the human's attention.

To obtain a software agent for these purposes, four models have been designed that are aimed at manipulating a person attention at a specific location: (1) a dynamical system model for attention, (2) a reasoning model to generate beliefs about attentional states using the attention model for forward simulation, (3) a discrepancy assessment model, and (4) a decision reasoning model, again using the attention model, this time for backward desire propagation. The first two models were adopted from earlier work [5].

The ambient agent has been implemented in a case study where participants perform a simplified version of the Tactical Picture Compilation Task. Within this case study an experiment was conducted to validate the agent's manipulation. The participants, both in the experiment discussed in this paper as well in earlier pilot studies, reported to be confident that the agent's manipulation indeed is helpful. The results of the validation study with respect to performance improvement have also been positive.

Further investigation has to be done in order to rule out any order effects, which suggests more research with more participants. It is also expected that future improvements of the agent's submodels, based on the gained knowledge from automated verification will also contribute to the improved success of such validation experiments.

Finally, a detailed analysis and verification of the behaviour of the agent also provided positive results. Traces of the experiment were checked to see whether the agent was able to adapt the features of objects in such a way that they attracted human attention. Results show that when there was a discrepancy between the prescriptive and the descriptive model of attention, the agent indeed was able to attract the human's attention.

References

[1] Baron-Cohen, S. Mindblindness: an essay on autism and theory of mind. MIT Press, 1995.

[2] Bosse, T., Both, F., Gerritsen, C., Hoogendoorn, M., and Treur, J. Model-Based Reasoning Methods within an Ambient Intelligent Agent Model. In: M. Mühlhäuser et al. (eds.), Constructing Ambient Intelligence: AmI-07 Workshops Proceedings. LNCS, vol. 11, Springer Verlag, 2008, pp. 352-370.

[3] Bosse, T., Jonker, C.M., Meij, L. van der, Sharpanskykh, A., and Treur, J., Specification and Verification of Dynamics in Agent Models. *International Journal of Cooperative Information Systems*, vol. 18, 2009, pp. 167 - 193. Shorter version in: Nishida, T. et al. (eds.), Proc. of the Sixth Int. Conference on Intelligent Agent Technology, IAT'06. IEEE Computer Society Press, 2006, pp. 247-254.

[4] Bosse, T., Memon, Z.A., and Treur, J., A Two-Level BDI-Agent Model for Theory of Mind and its Use in Social Manipulation. In: Proceedings of the AISB 2007 Workshop on Mindful Environments, 2007, pp 335-342.

[5] Bosse, T., Maanen, P.-P. van, Treur, J., Simulation and Formal Analysis of Visual Attention, Web Intelligence and Agent Systems Journal, vol. 7, 2009, pp. 89-105.

[6] Heuvelink, A., Both, F.: Boa: A cognitive tactical picture compilation agent. In: Proc. of the Int. Conf. on Intelligent Agent Technology, IAT '07, IEEE Comp. Soc. Press, 2007.

[7] Itti, L. and Koch, C., Computational Modeling of Visual Attention, Nature Reviews Neuroscience, Vol. 2, No. 3, 2001, pp. 194-203.

[8] Levitt, J.B., and Lund, J.S. Contrast dependence of contextual effects in primate visual cortex. Nature, 387, 1997, 73-76.

[9] Marsella, S.C., Pynadath, D.V., and Read, S.J., PsychSim: Agent-based modeling of social interaction and influence. In: Lovett, M., Schunn, C.D., Lebiere, C., and Munro, P. (eds.), Proc. of the Int. C. on Cognitive Modeling, ICCM 2004, pp. 243-248 Pittsburg, Pensylvania, USA.

[10] Memon, Z.A., and Treur, J., Cognitive and Biological Agent Models for Emotion Reading. In: Jain, L., Gini, M., Faltings, B.B., Terano, T., Zhang, C., Cercone, N., Cao, L. (eds.), *Proceedings of the 8th IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT'08*. IEEE Computer Society Press, 2008, pp. 308-313.

[11] Nothdurft, H. Salience from feature contrast: additivity across dimensions. Vision Research 40, 2000, 1183-1201.

[12] Parkurst, D., Law, K., and Niebur, E. Modeling the role of salience in the allocation of overt visual attention. Vision Research, 42, 2002, 107-123.

[13] Posner, M. E., Orienting of attention, Q. J. Exp. Psychol., 32, 1980, pp. 3-25.

[14] Theeuwes, J., Endogenous and exogenous control of visual selection, Perception, 23, 1994, pp. 429-440.

[15] Theeuwes, J. Abrupt luminance change pops out; abrupt color change does not, Perception & Psychophysics, 57(5), 1995, 637-644.

[16] Treisman, A. Features and objects: The 14th Bartlett memorial lecture. Q.J. Experimental Psychology A. 40, 201-237.

[17] Turatto, M., and Galfano, G., Color, form and luminance capture attention in visual search. Vision Research, 40, 2000, 1639-1643.

[18] http://www.tobii.com.

[19] http://www.yoyogames.com/gamemaker.