Prediction Of
Saccadic Eye Movements With Dynamic Scenes

Diploma Thesis
by
Jan Drewes

Submitted at the
University of Lübeck

Institute for Neuro- and Bioinformatics

Lübeck, Germany

June 2003
Prediction Of
Saccadic Eye Movements With Dynamic Scenes

Diplomarbeit
im Rahmen des Informatik-Hauptstudiums

Vorgelegt von
Herrn Jan Drewes

Ausgegeben von
Herrn Prof. Dr. rer. nat. Thomas Martinetz
Institut für Neuro- und Bioinformatik

Betreut von
Herrn Dr.-Ing. Erhardt Barth
Institut für Neuro- und Bioinformatik

Lübeck, Juni 2003
Statement of Originality

The work presented in this thesis is, to the best of my knowledge and belief, original, except as acknowledged in the text. The material has not been submitted, either in whole or in part, for a degree at this or any other university.

Lübeck, June 2003

(Jan Drewes)
Abstract

Today's visual communication systems are based on primary image features like color, brightness and contrast; these systems are unaware of the actual message that is to be conveyed by an image. The information that is perceived depends not only on the primary image features, but also on the way an observer looks at the image. This “way of looking”, the scan path, is an image attribute that is just as important for the reception of the image’s message as are the primary image features. To enhance tomorrow's visual communication systems, we want to transfer a given scan path onto another person. In order to be able to do this, we will need to better understand the mechanisms involved in choosing a gaze target, with the goal of being able to record, process and display a scan path along with the more common image attributes. As a step in this process, this thesis presents an algorithm capable of predicting eye movements of an observer viewing a dynamic scene based on a history of locations attended to in the past and also salient features of the current image. Salient features are computed based on the intrinsic dimension of the displayed video sequences. A mechanism to dynamically adapt this algorithm to an individual observer is presented.

This thesis is part of the Itap efforts at the Institute of Neuro- and Bioinformatics of the University of Lübeck, which is again part of the interdisciplinary project “ModKog”, supported by the German Federal Ministry of Education and Research.
# 0 Contents

0 Contents.......................................................................................................................................1
1 Introduction...................................................................................................................................2
   1.1 The human eye....................................................................................................................2
   1.2 Foveated vision and scan path...........................................................................................4
   1.3 Information technology for active perception: Itap............................................................5
2 Models of eye movements and attention..................................................................................6
3 Predicting eye movements.........................................................................................................8
   3.1 Scan path decomposition: fixation, tracking, saccades....................................................8
   3.2 Dealing with dynamic scenes...........................................................................................10
      3.2.1 Dynamic saliency.......................................................................................................10
      3.2.2 Wanted: An alphabet of changes.............................................................................12
      3.2.3 Intrinsic dimension and structure tensor.................................................................12
      3.2.4 The candidate list......................................................................................................14
   3.3 A reference to compare with: M1....................................................................................15
   3.4 Predictions based on previously attended locations: M2..............................................16
   3.5 The learning procedure for M2........................................................................................17
   3.6 Predictions based on salient features: M3......................................................................18
   3.7 The learning procedure for M3........................................................................................19
   3.8 Algorithm summary..........................................................................................................20
4 Experimentation and evaluation.............................................................................................21
   4.1 Video Sequences...............................................................................................................22
      4.1.1 Format and viewing size...........................................................................................22
      4.1.2 Content description...................................................................................................23
   4.2 Eye Movement Recordings...............................................................................................26
   4.3 Implementation.................................................................................................................27
   4.4 Problems and Pitfalls........................................................................................................28
   4.5 Computational results......................................................................................................30
      4.5.1 Sequence no. 1 “SimpleSequence”..........................................................................30
      4.5.2 Sequence no. 2 “AldiKreuzung”...............................................................................39
5 Discussion and Outlook............................................................................................................42
6 Mathematical addendum.........................................................................................................43
7 Image Listing ............................................................................................................................46
8 Bibliography..............................................................................................................................48
9 Acknowledgement....................................................................................................................49
1 Introduction

1.1 The human eye

Of all the human senses, the eye is probably the most sophisticated sensor. Light enters the human eye through the frontal optical opening, the pupil. Passing through the lens, the light is focused and projected to the rear wall of the eye, the retina. The retina is a complex network of highly sophisticated cells with the purpose of receiving and preprocessing the projected image.

The image recording capability of a human eye is mainly implemented through two types of light sensitive cells, the rods and the cones. The rods are responsible for sensing brightness and have greater sensitivity, while the cones are responsible for detecting different wavelengths (colors). The rods are the dominant sensor type for night vision, while the cones are the dominant sensors for day vision. This is why “at night, every cat is gray” - we can only reliably determine a cat’s fur color when there is enough light for the cones to detect different wavelengths. To do this, there are 3 subtypes of cones, sensitive to red, green, and blue light. Though each type of cone has an area of peak sensitivity, they are also capable of detecting the wavelengths surrounding their respective peak area. The color/wavelength actually seen is computed through adding/subtracting the output of several different cones. For example, the color yellow would be “seen” when both red - and green – sensitive cones are stimulated equally while blue cones are not or barely stimulated, and white would mean equal stimulation of all 3 cone subtypes.
The two types of receptors are not distributed equally over the retina, instead there is a high concentration of cones around the center of the retina, the fovea, consisting of about 200000+ cones (literature differs here, see e. g. [1]) and almost zero rods. From the fovea outwards, cones are rapidly reduced in numbers down to a relatively even spreading, while rods start to increase in numbers up to a peak density at about 18° from the center of view, from where their density is gradually lowered towards the outer regions. Due to this, the fovea is the area of the retina where the most image details can be detected, and where colors can be registered with the highest accuracy. At the outer regions, image resolution is much lower, especially color resolution. For a comprehensible and more detailed introduction to the human eye, see, e. g., [1].

From the field of view, the fovea spans only about 1-2°. Caused by this design, image recording with a human eye differs significantly from, for example, taking a photo. It is believed that nature created this design to limit the amount of information that needs to be transferred through the bottleneck of the optical nerve to the brain to be processed there – we see with our brains, our eyes merely collect the necessary data to do so (though there is a certain amount of preprocessing being done in the retina). Since we cannot register an entire scene in parallel, a serial strategy has evolved to do the job.
1.2 Foveated vision and scan path

Because of the foveated nature of human gaze, it is not possible to “see” an entire scene with one look. Instead, we need to direct our gaze to the important locations of what there is to be seen – we need to “scan” the scene to enable our brain to puzzle together a mental image resembling the “outside” world. In other words, as shown repeatedly, vision is a highly active process ([5],[6],[7]).

The pattern that we employ while doing this is thus called the “scan path”. The way we scan what we see has a major influence on the way we interpret and process visual stimuli, and is known to show large inter-subject differences.

The scan path has already been shown to be useful for object recognition [8] and video compression [9]. As a logically consistent extension of previous work the scan path could be used to improve vision-based communication and behavior as proposed in [10],[3].
1.3 Information technology for active perception: Itap

This thesis is part of the research efforts that aim at Information technology for active perception (Itap\(^1\))[3], being conducted at the Institute for Neuro- and Bioinformatics of the University of Lübeck.

As of today, visual communication systems are based on primary image features like color, brightness and contrast; these systems are unaware of the actual message that is to be conveyed by an image. The main idea behind Itap is that the way we look at a visual input (mostly the scan path) is an image attribute that is just as important for the reception of the image's message as are the primary image features (see image 7, page 4). If we want to transfer this way of looking onto another person, it will be necessary to record, process, and display the scan path along with the ordinary image data. To “record” a scan path, one can employ eye-tracking equipment – systems with sufficient precision and resolution are recently becoming commercially available, though they still show some deficiencies, mainly regarding restrictions to the subjects mobility and ease-of-use issues. To “process” a scan path could mean to re-adjust a recorded scan path to a modified image content, or even to generate an artificial one that will fit the image's conveyed message best. “Displaying” will require scan path guidance; the viewer will need to be manipulated into following the predefined path, if possible, without requiring the viewer to make a conscious effort, and preferably even subliminally, without the viewer even noticing.

An important area of applications is that of augmented vision with scan path guidance as an interface. As an example one may imagine a common situation in everyday traffic; As we know, a person trying to cross a road is sometimes in danger of being overlooked by the driver of an approaching car, because of fading daylight, or due to some distraction. It could be very helpful if the car's systems were able to detect the driver's current focus of attention, and then (if needed) to subliminally direct the driver's gaze towards the pedestrian ahead. This technology will allow for a very natural, intuitive integration of many technological aids into everyday life.

Itap research is supported by the German Federal Ministry of Education and Research as part of the interdisciplinary project ModKog\(^2\).

---

1  Itap, http://www.inb.uni-luebeck.de/Itap

2  ModKog, http://www.inb.uni-luebeck.de/Itap/modkog.html
2 Models of eye movements and attention

In order to process and display scan paths, it will first be necessary to better understand the way humans design their individual scan paths. In recent years, many attempts have been made to create a model of the basic functions involved in eye movements and the direction of attention.

The relevant models that have been published usually divide the factors of attention control into bottom-up and top-down components. Though task-oriented motivations can sometimes bias the process of picking a visual target, the bottom-up aspect is mainly driven by scene contents and is usually involuntary, “automatic”, allowing for a seemingly effortless picking of (unspecific) salient targets to attend to.

The other part of attention control is the top-down aspect. This is the part where intention and conscious motivations as well as subconscious, yet task-related factors come to play, and where an effort has to be made by the subject to change his gaze target. Due to the highly complex processes involved in conscious thinking as well as the vast inter-subject differences, modeling the top-down aspect would require such an intimate knowledge of each observed individual that it appears to be hardly possible to create a working model involving all of the factors that drive a person's mind. The relevant models therefore agree on the point that they concentrate on modeling the bottom-up aspects of attention and include, if at all, only very basic elements of top-down influences [11].

The models seem to converge on computing a “saliency map” describing the saliency of every location of the image, thus modeling the attractiveness for bottom-up attention control. Typically, the attended location is then chosen by a winner-take-all strategy, directing attention to the most salient spot available on the saliency map. Another mechanism, called “inhibition of return”, suppresses the recently chosen locations, thereby enabling the strategy to chose the next-salient spot, and so on, until all or a certain number of salient regions have been analyzed.

L. Itti’s model of saliency computation

In Itti's model from 1996, the input image is decomposed through linear filtering into basic features (color, intensity and orientations), resembling some of the most prominent filtering also done in primate / human visual processing. These features are then transformed with a center-surround mechanism to form different feature maps, which are then again linearly combined into a main saliency map. From this map, a scan path is generated by choosing most salient locations through a winner-take-all strategy with inhibition of return[12].

Image 9: L. Itti's version of the computation of a saliency map[4]
So far, most of the published models deal with static scenes only; even though dynamic scenes are relevant for robotics and active vision [13], few authors have proposed models of attention with dynamic scenes [14],[15],[16] and no model that has been validated on a comprehensive set of dynamic and natural visual inputs has come to the author's attention. Partially this is because of a lack of data due to technological limitations.
3 Predicting eye movements

Understanding attention control would greatly benefit from understanding saliency as it is perceived by the human eye. To achieve this, one could try to predict a person's scan path on a given image sequence with the help of some computed saliency map, and then learn from errors made (deviations from the subject's actual gaze positions). Before we can start to work with actual scan paths, we need to take a closer look at the major components of a typical scan path.

3.1 Scan path decomposition: fixation, tracking, saccades

The major components of a scan path that we will refer to in this context are fixation, tracking, and saccades. Though there exists a number of additional factors like rotation or nystagmus, they are either excluded from our study by choice of subjects or they will have no significant effect on our results (e. g., eye rotation will not interfere with gaze detection).

Fixations describe how the eye focuses on a spot that is static relative to its observer. In addition to the factors mentioned above, there are always different kinds of micro-movements and random noise involved in all kinds of eye activity, yet they are very small and can be ignored for our purpose. Therefore, we will treat fixation as zero eye movement.

Tracking (Pursuit) is what we do when we are focusing on a moving object. It does not matter if the object is moving on a straight line, in a circle or in weird patterns, as long as the object's motions can be steadily followed by the observer's gaze. Tracking can appear to be a fixation plus a shift that corresponds more or less directly to the motion of at least one of the current scene's features. Of course, tracking will only occur in dynamic scenes since static images have per definition no moving objects. Since we will be dealing with dynamic scenes in the form of frame-based image sequences, it is important that both video frame rate and per-frame movement of the object will allow for a continuous motion appearance. This is taken care of by choosing an appropriate media type and an according display program (see chapters 4.1 to 4.4).

Saccades are the movements our eyes perform to shift our gaze from one point of interest to another or to compensate for object movements that we were unable to track (see above). This may include attending to different features of the same structure (e. g. different corners of the same square). Saccades are the most rapid eye movements, in fact, their angular speed and acceleration are limited only by the eye's inertia and its muscle's capabilities; their speed can be as high as 200 – 400°/s [17]. Also, saccades are ballistic – once a saccade has been triggered, their target cannot be altered. If a targeted object suddenly moves, or if a new target is desired to be attended during the execution of a saccade, then the current operation must be completed before a new saccade can take place to correct the gaze position. Additionally, visual input is almost completely suppressed during a saccade, rendering humans blind to changes in the image for the duration of the saccade. This is why one never really sees his eyes moving when looking into a mirror.
The three basic components can be sequentially combined in almost any fashion, but quite obviously they cannot happen simultaneously.
3.2 Dealing with dynamic scenes

In this thesis the temporal aspects (motion and image changes) will be of significant importance to the computation of our saliency map, as we will be dealing with dynamic scenes.

We deal with dynamic scenes because non-static imaging is an important aspect of visual communication systems (Itap involves gaze contingent displays). We believe that when viewing static images, top-down as well as random aspects will dominate after a few moments because there is nothing “new” to see and the observers gaze starts wandering around, examining relatively unimportant image features (e.g., the artist's brush strokes).

When dealing with dynamic scenes with a certain minimum of changes over time, we expect the bottom-up component of attention guidance to be in a more dominant role, allowing for a more efficient prediction. Since we incorporate dynamic scenes, we will of course have dynamic saliency. We will include temporal aspects not just by computing saliency per each 2D-frame in time, but also by including temporal image features like motion and scene changes in each respective saliency map.

3.2.1 Dynamic saliency

It is known that when viewing static images, the eye's attention is attracted by high-contrast image regions like edges and corners, that can be interpreted as local changes of the image's intensity function \( f(x, y) \). It has been proved that the regions of an image, where the intensity function shows significant changes (i2D regions\(^3\)) uniquely specify an image [18].

![Image 10: A box](image10.png) ![Image 11: Boxes i1D features (symbolic)](image11.png) ![Image 12: Boxes i2D features (symbolic)](image12.png)

For example see the image to the left; if we know its i1D features (displayed symbolically in the middle image), we are able to reconstruct the entire image from only the edges and possibly one reference color sample; this is also possible using only the image's i2D features displayed in the image to the right. This demonstrates that where there are no changes in the image, no information can be conveyed. In dynamic scenes, similar effects appear: Attention is less likely to be directed towards a region of uniform intensity that does not change in time, as there appears “nothing to be seen”. On the other hand, areas with high

\(^3\) i2D regions are locations in an image, where the intrinsic dimension is at least 2. See also chapter 3.2.3
contrast and/or motion will attract a person's attention. In other words, the system is sensitive to changes. But what type of changes?
3.2.2 Wanted: An alphabet of changes

We would like to have a basic set of change types where each change type has an individual amount of saliency assigned. Such a set could be considered an alphabet from which all kinds of changes could be composed, and into which all changes could again be decomposed, revealing their primary change types and allowing to sort them in their order of saliency. This issue has been analyzed from different viewpoints with no all-embracing, non-ambiguous alphabet found; still, we believe that there is such a basic alphabet that should be taken into consideration and has not been adequately covered in the literature.

We will base our approach to the creation of a saliency map on the concept of intrinsic dimensionality that has been introduced for images in [19] and shown to be useful for modeling attention with static images in [20]. Here, we will use the local intrinsic dimension of each location in every image of a sequence as an indicator for the degree of saliency. As we are using the intrinsic dimensionality, we are making implicit use of the alphabet mentioned above, yet we do not attempt to distinguish the individual change types.

3.2.3 Intrinsic dimension and structure tensor

Our image sequence will be represented by its intensity function $f(x,y,t)$, which assigns one value of intensity to each pixel of our image (encoded through $x,y$) for each sequence frame (encoded through $t$). As for this thesis we will be dealing with monochrome images only.

The intrinsic dimension of a pixel in an image sequence can be understood as the “changyness” of the pixel relative to its immediate neighborhood. If we see the pixel as embedded in 2 spatial and one temporal dimensions $(x,y,t)$, then its intrinsic dimension increases with every non-constant dimension. A non-constant dimension could e. g. be an edge in an image, or, on a per-pixel-base, a neighboring pixel with a different color/intensity. Note here that where dynamic scenes or image sequences are treated, there is a spatial neighborhood as well as a temporal neighborhood to be considered.

A pixel's intrinsic dimension (“inD”, where $n$ represents the dimensionality) is

0 if the signal is constant in all directions $\{f(x,y,t) = c\}$ \hspace{1cm} i0D
1 if the signal is constant in only 2 directions $\{f(x,y,t) = g(u)\}$ \hspace{1cm} i1D
2 if the signal is constant in only 1 direction $\{f(x,y,t) = g(u,v)\}$ \hspace{1cm} i2D
and it is 3 if there is no constant direction.

The evaluation of the intrinsic dimension is here implemented by using the structure tensor $J$, which is well known in the computer-vision literature.
Based on the image intensity function \( f(x, y, t) \) the structure tensor is:

\[
J = w^* \begin{pmatrix}
  f_x^2 & f_x f_y & f_x f_t \\
  f_x f_y & f_y^2 & f_y f_t \\
  f_x f_t & f_y f_t & f_t^2
\end{pmatrix}
\]

(see chapter 6)

where subscripts indicate partial derivatives and \( w \) is a spatial smoothing kernel that is applied to the products of first-order derivatives. Smoothing is necessary especially when dealing with natural scenes, as video recordings usually contain a certain amount of noise, which would otherwise result in large amounts of “bogus” saliency all over the image. Due to the nature of the structure tensor, the intrinsic dimension of \( f \) is zero if the eigenvalues of \( J \) are all zero, and in general it is \( n \) if \( n \) eigenvalues are different from zero. However, we do not perform the eigenvalue analysis of \( J \) since it is possible to derive the intrinsic dimension from the invariants of \( J \), which are:

\[
H = \frac{1}{3} \text{trace}(J) = \lambda_1 + \lambda_2 + \lambda_3
\]

\[
S = M_{11} + M_{22} + M_{33} = \lambda_1 \lambda_2 + \lambda_2 \lambda_3 + \lambda_1 \lambda_3
\]

\[
K = \text{det}(J) = \lambda_1 \lambda_2 \lambda_3
\]

where \( M_{ij} \) are the minors of \( J \) obtained by eliminating the row \( 4-i \) and the column \( 4-j \) of \( J \). The \( \lambda_k \) are the eigenvalues of \( J \). Note, however, that we do not need to estimate these eigenvalues, as there are alternative methods for each of the three invariants. Since \( J \) is a positive definite matrix, the intrinsic dimension is at least 1 if \( H \) differs from zero, at least 2 if \( S \) differs from zero, and at least 3 if \( K \) differs from zero.

\[
H \neq 0 \quad \rightarrow \quad i1D
\]

\[
S \neq 0 \quad \rightarrow \quad i2D
\]

\[
K \neq 0 \quad \rightarrow \quad i3D
\]

Examples of the generated filter images for \( S \) and \( K \) can be seen in chapter 4.1.
Our current implementation is limited to using only the invariant $S$ for saliency map creation. This seems to be the obvious choice because a value of $S \neq 0$ indicates an intrinsic dimension of at least 2 and, therefore, suppresses regions of dimension less than 2, which are redundant. The $S(x, y, t)$ values are then used to obtain a list of candidate locations as follows.

Regions with $S(x, y, t) < \theta$ are ignored. The threshold value $\theta$ is given in percent relative to the maximum present saliency computed on normalized images and remains a parameter of the model. Connected regions with $S$ values above the threshold are then reduced to only one location, anchored at the highest $S$ value of the region (when there are more than 1 equally valued peak locations, the one to the top-left is chosen due to the raster-scanning pattern of the connected component analysis). This location is then written to a list of candidate locations $X^cand_i = (x^cand_i, y^cand_i)$ with $i = 1, \ldots, L$, ordered by the maximum and the mean values of $S$ in that region.

In image 13 an example with a short (7 frame) sequence can be seen. In frame 0, the scene is uniformly white; in frame 1, a black box pops into existence. Immediately, the entire boxes border becomes 2D and can be found in the saliency map (based on $S$). Due to the Gaussian smoothing, the $S$ values show softer contrast than the original box. Frame 6 shows the situation for a stationary box after popping into the scene; as we are also employing a Gaussian temporal smoothing with radius 5 (for more parameters, see chapter 4) to eliminate noise, frames 2 to 5 are gradually blending over from frame 1 to frame 6. On the right of image 13 the resulting candidate points are displayed. For frames 1 – 4, the entire saliency structure is above the chosen $\theta$ (which is usually around 50%), resulting in only one connected region and only one candidate point. From frame 5 on, the more prominent corners are still above the threshold, while the connecting edges are being cut out, resulting in 4 individual connected regions and 4 candidate points, one at each corner of the original box. Since the box is symmetric and immobile, all 4 points will be valued with the same weight.

Please note that above image has been inverted for better printing; in the oncoming context, low saliency will be displayed as black, high saliency as white.

We compute saliency on different scale levels forming a pyramid, using both spatial and temporal down-sampling with Gaussian filtering. The number of pyramid levels $\Lambda$ can be changed, but was fixed to $\Lambda = 4$ here.

Sample saliency-images computed on synthetic as well as natural scenes can be found in chapter 4.
3.3 A reference to compare with: M1

As a reference model, we will not use an imaginary ideal (zero error) prediction, as we believe that to be unrealistic. Instead, we will compare our results with a fairly simple “prediction”-model:

M1: \( X_{i}^{pred} = X_{i-1}^{ref} \)

This model is strongly related to fixations (see chapter 3.1). Whenever our subject truly fixates an object, M1 will be right on. Of course, whenever our subject is doing anything else than fixation, M1 will always be off-target by exactly the gaze movement between frame \( t \) and \( t-1 \). Accordingly, our mean cumulative quadratic error function for M1 is

\[
E_{M1} = \frac{1}{T-1} \sum_{i=1}^{T-1} \left( X_{i}^{ref} - X_{i-1}^{pred} \right)^2
\]

and if we plot the errors actually made by this model, we will see a curve identical to the one we will achieve when plotting a subject’s gaze shift pattern.

Absolute error and error histogram plots of natural and synthetic image sequences can be found in chapters 4.5.1 and 4.5.2.
3.4 Predictions based on previously attended locations: M2

Our first attempt to improve predictions will use previously attended gaze positions to form a learning, adaptive mechanism which is designed to be able to predict (short-term) trends. This will be useful for tracking and pursuit of objects.

By recording the subject’s eye movements with our eye tracker, we know the history of locations the subject has previously attended. We use supervised-learning techniques to find the best possible linear prediction one can make based on a history of size $N$. The predicted location is defined by:

$$\text{M2}: \quad X_{t}^{\text{pred}} = X_{t-1}^{\text{ref}} + A_{t-1} \times P_{t-1}$$

Where $X_{t}^{\text{pred}} = (x_{t}^{\text{pred}}, y_{t}^{\text{pred}})$ is the vector-valued location predicted for the current frame and $X_{t-1}^{\text{ref}} = (x_{t-1}^{\text{ref}}, y_{t-1}^{\text{ref}})$ is the real (“reference”) gaze position of the previous frame. So, we are expressing the next attended location as a shift relative to the last known location.

$P_{t-1} = (X_{t-2} - X_{t-1}^{\text{ref}}, X_{t-3} - X_{t-2}^{\text{ref}}, \ldots, X_{t-N} - X_{t-N}^{\text{ref}})$ is an array of the $N$ most recent shift vectors expressed relative to the last currently known location $X_{t-1}^{\text{ref}}$.

The $N \times 2$ matrix $P_{t-1}$ is mapped by the $1 \times N$ matrix $A_{t-1}$ to a $2 \times 1$ displacement vector that defines the shift of attention from the previous to the current frame. The matrix $A_{t-1}$ is determined by supervised learning and is updated continuously after every prediction, using the reference data provided by the eye-tracker.

M2 is expected to behave somewhat like an adaptive temporal filter, capable of following (and thus predicting) constant short-term elements in the subject's gaze – mainly fixations and trackings / pursuits, but saccades may be at least partially predicted when their direction conforms with the general trend over the last few frames or when they happen at the very border between two frames. In the latter case, a part of the saccade will occur in the previous frame, causing the matrix to learn something out of the sudden increase in gaze shift distance, thus allowing for a more fitting estimation of the next gaze position (the second part of the saccade). Of course M2 will never be able to completely predict a saccade, as the temporal extrapolation cannot account for sudden changes in the subjects' gaze pattern.
3.5 The learning procedure for M2

In M2, during the duration of a test run, the matrix $A$ is not constant; instead, as already mentioned, it will be constantly updated after every prediction attempt. The learning rule is incremental and minimizes by gradient descent the mean prediction error, which is defined as the sum of quadratic differences between the predicted and the actually attended locations:

$$E_{M2} = \frac{1}{T-1} \sum_{i=2}^{T-1} \left( X_i^{ref} - \left( X_{i-1}^{ref} + A_{i-1} \times P_{i-1} \right) \right)^2$$

This error can be minimized by an iterative procedure, i.e., an incremental learning strategy, by using the following update rule [21]:

$$A_{i} = A_{i-1} + \alpha \cdot e \cdot P_{i-1}^T$$

where $\alpha$ is the learning rate and $e = X_i^{ref} - \left( X_{i-1}^{ref} + A_{i-1} \times P_{i-1} \right)$ the current error that is used for incremental learning. The learning rate is the distance by which the algorithm walks down the error function in the direction of the gradient $e \cdot P_{i-1}^T$. We have experimented with different constant learning rates and also with rates that have been decremented exponentially. Best results, however, have been obtained with a procedure that estimates the optimal learning rate at each iteration step and then weights this value with a constant value $\alpha$. In this case the learning rate depends on the current error and is defined by $e = \frac{e \cdot P^T \cdot e^T}{P^T \cdot e^T}$. This expression is found by a line-search method that minimizes the error on the current input, see chapter 6.
3.6 Predictions based on salient features: M3

As shown in our results (see chapter 4), the temporal predictions (M2) are good with respect to the mean prediction error. However, the prediction is based on previously attended locations only and can therefore not predict sudden shifts in attention (saccades) that are triggered by, for example, the sudden appearance of a novel object. In other words, M2 does not include saliency information. We therefore extend our model M2 to predict the next gaze location by including the computed saliency / candidate points:

\[
X_{i}^{\text{pred}} = X_{i}^{\text{ref}} + A_{i-1} \times P_{i-1} + B_{i-1} \times S_{i}
\]

The array \( S_{i} \) holds a number of salient candidate locations that are extracted from each pyramid level of the current frame at time \( t \) and optionally also salient locations extracted from previous frames up to a history of \( M \) frames. The index \( i = 1, \ldots, L \) is used to denote a number of up to \( L \) salient locations estimated for every frame on every pyramid scale level. Thus, \( S_{i} \) holds \( M \times L \times A \) locations in a \( M \times L \times A \times 2 \) matrix. The procedure for obtaining the salient locations is as described above.

You may want to pay attention to the fact that it is actually \( S_{i} \) and not \( S_{i-1} \). This expresses the fact that we are indeed processing data from the current image for which we have no reference data (yet). Still, this information is not coming out of the (very near) “future”; instead, we are able to compete with and possibly even outrun the subjects saliency registration process. We may do our processing here even before the subject actually attends to the contents of this particular image. If, in some near or distant future, we would decide to modify the saliency of a given scene, this would be just the place to do it if we needed to deal with real-time imaging processes.

The \( 1 \times M \times L \times A \) matrix \( B_{i-1} \) maps all the salient locations to a displacement vector that defines the saliency-based contribution to the shift of attention from the previous to the current frame. The actual shift is the sum of the saliency-based and the temporal contribution to the prediction. We obtain the matrix \( B_{i-1} \) by using the same learning procedure as for the matrix \( A_{i-1} \). Note, however, that the matrices \( A_{i-1} \) and \( B_{i-1} \) are now learned simultaneously, i.e., the general prediction error used to drive the learning procedure is obtained from both matrices.

Of course, one could use only one matrix for both the previously attended and the saliency-based locations and only one matrix for the mapping. Nevertheless, the separation seems useful for conceptual reasons and because it is now possible to update the matrices \( A_{i-1} \) and \( B_{i-1} \) at different rates. Intuitively, the matrix \( A_{i-1} \) would learn to track and make short-time predictions, and the matrix \( B_{i-1} \) would rather learn a strategy for choosing a location from a list of candidate locations. It thus seems reasonable to use different learning rates.
3.7 The learning procedure for M3

The learning rule for M3 is analogous to the one for M2. Each prediction returns an error value \( e \); the difference is that we now need to split the error for the individual learning procedures of the matrices \( A \) and \( B \). We do this by introducing the error weights \( y^A \) for the temporal matrix \( A \) and \( y^B \) for the saliency matrix \( B \).

Now, based on M3’s error function:

\[
E_{M3} = \frac{1}{T-1} \sum_{i=2}^{T-1} \left( X_i^{ref} - \left( X_{i-1}^{ref} + A_{i-1} \times P_{i-1} + B_{i-1} \times S_{i} \right) \right)^2
\]

we adapt \( A \) and \( B \) with the update procedure for M2, but with the following changes:

\[
A_t = A_{i-1} + \alpha (e y^A P_{i-1}^T) \quad \text{and} \quad B_t = B_{i-1} + \alpha (e y^B S_t^T)
\]

where \( \alpha \) are again the learning rates as defined in chapter 3.5 and \( e = X_t^{ref} - \left( X_{i-1}^{ref} + A_{i-1} \times P_{i-1} + B_{i-1} \times S_t \right) \) is the current error that is used for incremental learning. Though it seems logical to constrain our error weights to \( y^A + y^B = 1 \) with \( y^A, y^B \in [0,1] \), this is not really necessary. For example, \( y^A + y^B = 2 \) would have the same effect as a doubled learning rate \( \alpha \) with \( y^A + y^B = 1 \).

---

Image 14: The learning mechanism
3.8 Algorithm summary

I. Prior to the actual processing, the original scenes are shown to the test subject. The test subject's gaze data is recorded.

II. The original image sequence is transformed into a sequence of filter images based on the invariant S of the structure tensor J.

III. A list of candidate locations is assembled from the results of a connected component analysis for each filter image.

IV. The computed data and the recorded gaze data enter the adaptive prediction algorithm. Matrices A and B are learned incrementally.

V. Predictions are made according to the models M1, M2 and M3. The resulting prediction errors are stored and then analyzed manually.

*Image 15: Algorithm summary*
4 Experimentation and evaluation

For performance evaluation, we trained and tested the models with our own recordings of eye-movements. We present results in terms of the prediction errors. The parameters of the model are the following:

The derivatives have been computed by finite differences after spatio-temporal Gaussian low-pass filtering with a kernel of size 5x5x5 and \( \sigma_1 = 3 \) for all variables \( (x, y, t) \).

The kernel \( w \) that convolves the product terms of the structure tensor \( J \) was the same as the one used for estimating the derivatives \( (\sigma_2 = 3) \). The influence of \( \sigma_1 \) and \( \sigma_2 \) on the prediction errors has not been analyzed yet, but is not expected to be significant.

The threshold \( \theta \) was adaptive and set to 0.5 times the maximum of the current frame. For the results presented here we used a minimal configuration with \( N = 2 \), \( M = 1 \) and \( L = 4 \). The four saliency locations have been obtained by choosing only one location from each scale of the Gaussian \( S \) pyramid. The learning rate was scaled by \( \alpha = 0.001 \) in order to prevent an excessive learning step size.
4.1 Video Sequences

4.1.1 Format and viewing size

The results presented here have been obtained with the following video sequences. Videos were recorded in DV PAL format (720x576, 24bit color, 50 half-frames per second) with a Sony DCR PC1 digital camcorder using miniDV digital tapes. After transferring the videos to a host PC they were transformed to ¼ PAL resolution (360x288) with 8bit grayscale color space, 25 progressive frames per second and stored as a sequence of JPEG images for display on PC2 as well as a sequence of BMP images for processing on PC1. PC2 (see chapter 4.2) was used to display the sequences on a Sony Multiscan E500 21” CRT Monitor at a screen resolution of 800x600, zoomed to full screen, with a screen refresh rate of 75Hz. The actual image size during display was approximately 40 x 30cm, viewed from a distance of 75cm, thus spanning a field of view of about 30°.
4.1.2 Content description

**Sequence No.: 1 “SimpleSequence”**

This sequence is purely synthetic, showing a number of squares (max. 2 at the same time) at various maneuvers like popping/blending in and out, moving horizontally and diagonally, or just sitting there. The virtual camera position was fixed.

![Sample frame no. 300 from "SimpleSequence"](Image 16)

![S-valued saliency for frame no. 300 (SimpleSequence)](Image 17)

![K-valued saliency for frame no. 300 (SimpleSequence)](Image 18)

The shown example frame (images 16,17,18) shows 2 squares, the one in the top-right corner sitting still, while the second one is moving diagonally from top-left to bottom-right. The dark spots in the corners of the salient structures symbolize the computed candidate points.

**Event list for “SimpleSequence”**

<table>
<thead>
<tr>
<th>Frames</th>
<th>Event description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-37</td>
<td>White square no.1 appears and stays at top right without motion</td>
</tr>
<tr>
<td>38</td>
<td>White square no. 1 pops out of scene</td>
</tr>
<tr>
<td>39-53</td>
<td>Blank screen</td>
</tr>
<tr>
<td>54</td>
<td>White square no. 2 pops in at top left and starts moving diagonally towards the bottom right</td>
</tr>
<tr>
<td>120</td>
<td>White square no. 2 arrives at bottom right and pops out again</td>
</tr>
<tr>
<td>121-135</td>
<td>Blank screen</td>
</tr>
<tr>
<td>136-162</td>
<td>White square no. 3 fades in at the top right of the scene</td>
</tr>
<tr>
<td>163-176</td>
<td>White square no. 3 stays at top right without motion</td>
</tr>
<tr>
<td>177-200</td>
<td>White square no. 3 fades out of the scene</td>
</tr>
<tr>
<td>201-217</td>
<td>Blank screen</td>
</tr>
<tr>
<td>218</td>
<td>Square no. 4 pops in to the scene a the top left and instantly starts moving down to the bottom right. Simultaneously, square no. 5 pops in to the scene at the top right and stays there</td>
</tr>
<tr>
<td>219-238</td>
<td>Square no. 4 moves towards the bottom right</td>
</tr>
<tr>
<td>284</td>
<td>Square no. 4 arrives at the bottom right, squares no. 4 and 5 pop out of the scene</td>
</tr>
<tr>
<td>285-299</td>
<td>Blank screen</td>
</tr>
<tr>
<td>300</td>
<td>Square no. 6 pops in to the scene a the top left and instantly starts moving down to the bottom right. Simultaneously, square no. 7 pops in to the scene at the top right and instantly starts moving towards the left</td>
</tr>
<tr>
<td>301-365</td>
<td>Square no. 6 moves towards the bottom right, Square no. 7 moves towards the left</td>
</tr>
<tr>
<td>366</td>
<td>Square no. 6 arrives at the bottom right, square no. 7 arrives at the left; squares no. 6 and 7 pop out of the scene</td>
</tr>
<tr>
<td>Frames</td>
<td>Event description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>367-381</td>
<td>Blank screen</td>
</tr>
<tr>
<td>382</td>
<td>Square no. 8 pops into the scene at the top left and starts accelerating towards the bottom right</td>
</tr>
<tr>
<td>383-447</td>
<td>After passing ½ of the distance, square 8 starts to decelerate, keeping it's orientation towards the bottom right</td>
</tr>
<tr>
<td>448</td>
<td>Square no. 8 comes to a stop at the bottom right of the screen, immediately popping out.</td>
</tr>
<tr>
<td>449-463</td>
<td>Blank screen</td>
</tr>
<tr>
<td>464</td>
<td>Square np. 9 pops in at the top left and immediately starts moving towards the bottom right</td>
</tr>
<tr>
<td>465-484</td>
<td>Square no. 9 keeps moving towards the bottom right</td>
</tr>
<tr>
<td>485</td>
<td>While Square no. 9 is still moving towards the bottom right, having passed about 1/3 of the distance, square no. 10 pops in at the top right and stays there.</td>
</tr>
<tr>
<td>486-513</td>
<td>Square no. 9 continues its movement towards the bottom right, while square no. 10 stays in it's position.</td>
</tr>
<tr>
<td>514</td>
<td>Square no. 9 continues its movement towards the bottom right, having passed about 2/3 of the distance. Square no. 10 pops out of the scene.</td>
</tr>
<tr>
<td>515-529</td>
<td>Square no. 9 continues its movement towards the bottom right.</td>
</tr>
<tr>
<td>530</td>
<td>Square no. 9 arrives at the bottom right and immediately pops out of the scene.</td>
</tr>
<tr>
<td>531-547</td>
<td>Blank screen</td>
</tr>
<tr>
<td>548-572</td>
<td>Square no. 11 starts to face in at the top left of the screen and accelerates towards the bottom right</td>
</tr>
<tr>
<td>573-584</td>
<td>Square no. 11 accelerates towards the bottom right of the scene; when passing ½ of the distance, it starts to decelerate, keeping its direction</td>
</tr>
<tr>
<td>585-610</td>
<td>Square no. 11 starts to fade out while decelerating, finally disappearing while simultaneously reaching the bottom right and a stop</td>
</tr>
<tr>
<td>611-627</td>
<td>Blank screen</td>
</tr>
<tr>
<td>671-634</td>
<td>Square no. 12 pops in to the scene a the top left and instantly starts moving down to the bottom right. Simultaneously, square no. 13 starts to fade in to the scene at the top right and stays there</td>
</tr>
<tr>
<td>635-667</td>
<td>Square no. 12 keeps moving towards the bottom right, Square no. 13 stays at the top right</td>
</tr>
<tr>
<td>668-693</td>
<td>Square no. 12 keeps moving towards the bottom right, Square no. 13 starts fading out</td>
</tr>
<tr>
<td>694</td>
<td>Square no 13 has disappeared, Square no. 12 reaches the bottom right and pops out of the scene</td>
</tr>
<tr>
<td>695-709</td>
<td>Blank screen</td>
</tr>
</tbody>
</table>

J. Drewes  Institute for Neuro- and Bioinformatics, University of Lübeck, 2003  24/49
Sequence No.: 2 “AldiKreuzung”

This was taken at a road crossing next to the author’s favorite discounter and shows a somewhat typical traffic scene with cars driving through from different angles as the traffic light cycles. The camera was positioned on a static mount next to a crossing with traffic lights.

The example shows some cars driving through from the right; the structures in the background, especially the trees, appear highly salient because they are bending in the wind and thus create a lot of motion. This is to be observed on the higher-frequency scales, though.
4.2 Eye Movement Recordings

Eye movements have been recorded using the commercial iViewX system produced by SMI⁴, that uses an infrared CCD camera pointed at the subject's eye as well as two infrared illumination sources. A chinrest was employed to minimize errors from the subject's head movements (see image 23). The camera image is recorded on PC number 1 using a common videograbbber card (conexant/brooktree based). Input images are being processed at 50 frames per second at ¼ PAL resolution, derived from individual half-frame images (the camera output is interlaced) through shrinking width by 50%. The gaze position is computed based on the position of the pupil, and the relative positions of the 2 corneal reflexes projected by the infrared illumination. Results are then scaled to the display resolution (800x600). Prior to each recording session the system was calibrated through a 9-point calibration routine provided by the iViewX client program “WinCAL” running on PC2. Also running on PC2 was our display program, capable of displaying sequences of JPEG-Images (and planned to be capable of displaying MPG videos as well). The use of a special viewing application was indicated by the need to synchronize data recording on PC1 with sequence displaying on PC2. A hardware synchronization is provided through PC2's parallel port and a digital I/O card monitored by the iViewX software on PC1. According to the manufacturer, the precision of the eye-tracker is around 0.5° in actually achievable conditions, resulting in about 60 discernible steps for the width of the field of view in our experiments.

---


J. Drewes  Institute for Neuro- and Bioinformatics, University of Lübeck, 2003
4.3 Implementation

The evaluation program consists of 2 main elements, both of which were programmed as console applications under Linux in C / C++ making extensive use of the Intel OpenCV and IPL\textsuperscript{5} libraries with emacs and GNU make.

The first element is a small suite of utilities that will accept a series of grayscale bitmaps and output one series of saliency images together with a series of candidate lists for each pyramid level. For our experiments, 4 pyramid levels were used, starting with the original \(\frac{1}{4}\) PAL (360x288@25Hz), then employing a (Gaussian) pyramidal downsizing in both spatial and temporal resolution to 180x144@12.5Hz, 90x72@6.25Hz and finally 45x36@3.125Hz.

The second element is the main program. It takes a series of saliency lists and a gaze data file as input. By exchanging the pre-recorded gaze data the test subject can be chosen; saliency needs to be computed only once per video sequence. When instructed to do so, the main program can also read original images and saliency images from all pyramid levels and combine them with gaze and prediction data to form up an explanatory image sequence as seen in the example below.

On the upper half of the image the computed saliency for the current frame can be seen for all four pyramid levels; already pointed out are the chosen candidate points, which can also be seen in the lower half of the image as colored dots (red, yellow, turquoise and blue, depending on the pyramid level), painted on the underlying original image, also in all four pyramid levels. The white, hollow target cross marks the actual gaze position of the subject, while the thin, yellow cross marks the predicted gaze position.

Statistical evaluations and image enhancements (for display) were made under Linux using Matlab and Octave/GNUplot as well as the GIMP. Video preprocessing (conversion of the original DV video into \(\frac{1}{4}\) PAL grayscale bitmap sequences and the preparation of the demonstration videos) was done under Windows 2000 and Windows 98 SE with Ulead Media Studio Pro 6.0.

\textsuperscript{5} IPL and OpenCV have now been joined into the latest version of OpenCV, which can be found at: www.intel.com/research/mrl/research/opencv/
4.4 Problems and Pitfalls

There are a number of things to be considered before looking at the results of our evaluations. The simplicity of the method used to create the list of candidate points which is then input into the learning procedure of M3 is a conceptual advantage due to its straight-forwardness; the downside is that it strongly relies on a correct measure to determine the most attractive salient spots. Currently, our measure only uses absolute values like average intensity and peak intensity; Other (relative) properties like proximity to other or previously attended locations is not included in our algorithm, even though it is quite possible that these properties have an effect on the process of choosing the next attended location. Also, our implementation of the structure tensor $\mathbf{J}$ weights spatial and temporal changes equally, yet it has not been proved that e. g. spatial motion is equivalent to purely temporal changes (like a moving square vs. a square blending in without actually moving around).

Furthermore, our algorithm M3 is a linear mapping: it includes the saliency part of the formula at every prediction cycle. Since humans don't usually perform more than 3 or 4 saccades per second, there will be no saccade to predict during at least 20 of the 25 frames per second. In a worst case scenario, this could mean that the algorithm will learn the saliency matrix $\mathbf{B}$ with plausible values 3 to 4 times out of 25, when gaze target changes occurred in relation to candidate locations, and with non plausible values, when saliency didn't matter, more than 20 times. Now, when the saliency matrix is learned with “non-plausible” values, this means that error minimization can only be done by approximating the individual coefficients to zero. In short, the saliency matrix will drift towards plausible, non zero coefficients at about 15% of the learning cycles, and it will drift towards zero-valued coefficients for about 85% of the learning cycles. This causes the coefficients of the saliency matrix to be much smaller than those of the temporal matrix at most times. To at least partially compensate for this, we increased the learning ratio of the saliency matrix by setting $y_B$ to 30 for “SimpleSequence” and to 5 for “AldiKreuzung” while leaving $y_A$ at 1 for all sequences.

Also, note that the main program currently needs 43 frames for initialization of internal buffers and other purposes, so the videos shown to the subjects were about 1,7 seconds longer than our plots. This has no influence on our learning algorithm.

Another point is that we do not implement inhibition of return in our process of choosing candidate locations. As we are dealing with dynamic scenes at a display ratio of 25Hz, this is probably unnecessary in a number of cases, like our “SimpleSequence”. In real life video sequences however, there may be a certain amount of static, “background” saliency, as can be seen with the trees in “AldiKreuzung”. Basically, they just wiggle a bit in the wind, thus maintaining a certain amount of limited, “stationary” motion and showing, against the white sky behind them, a fair amount of two-dimensional contrast. This is persistent through the entire duration of the scene, resulting in a number of candidate locations found among the trees; however, only the higher frequency / larger scale pyramid levels show these effects. Those locations may in fact be interesting to the observer at first, but, after a while, they will be ignored; they are still relatively salient, yet they do not “change” in the eye of the observer. This problem could be dealt with by implementing inhibition of return...
in our candidate point algorithm; to do this, some extra precautions will be necessary: The candidate point locations may be a little bit different from frame to frame, so one would have to track each individual candidate location to determine whether a salient spot is old and should therefore be inhibited, or whether it is new and should be included in the current saliency list. This is not yet implemented in our program.
### 4.5 Computational results

When invoked with a suitable set of parameters, the main program will output several files containing plain-text lists with the errors that occurred during prediction for further evaluation. We will show our prediction results in the form of these prediction errors.

#### 4.5.1 Sequence no. 1 “SimpleSequence”

The above image 25 shows the errors of our model M1 (identical to the distances of the subjects' gaze shifts) for the entire sequence no.1. The spikes can easily be identified as saccades, whereas the smaller and more noisy, but still well visible arches (e.g. from about frame no. 60 to about 120 or from about frame no. 230 to about 290) are trackings (pursuits) of diagonally moving squares. The relatively flat part before frame 50 is caused by the subject fixating the first (non-moving) square and the subsequent blank scene; occasionally, the subject would change from one corner to another within the same square resulting in small peaks. This recording matches well with the description of the happenings during the sequence (see scene event description, chapter 4.1.2).
Image 26: Prediction errors of M2 for "SimpleSequence"

Image 26 shows the prediction errors for our model M2 for the entire duration of the sequence. Clearly, the overall shape of the curve is very similar to the gaze shifts performed by the subject. When compared directly, one may notice that the arches appear a bit flatter than in the gaze plot (M1, image 25); this is due to the linear temporal prediction in our algorithm matching the subject's tracking speed. In the direct subtraction (M2 - M1, see Image 27) these prediction gains become more obvious (negative values mean M2 was better than M1).

Image 27: Difference of M2 and M1 for "SimpleSequence"
When comparing M3 with M1 ($M3 - M1$), see image 29), the results are somewhat similar to the comparison of M2 vs. M1 (negative means M3 was better than M1).

We notice a significant gain of M3 at around frame no. 100, which we did not see in the comparison of M2 vs. M1.
When looking at the histograms of the gaze shifts (images 30,31,32), one can recognize an increased quantity of smaller errors with a simultaneously decreased quantity of larger errors.
To compare M2 and M3 in a more explicative way, we will plot the cumulative errors of M2 and M3 in relation to those of M1; this means that M1 becomes our new reference.

As we are plotting cumulative sums of relative errors, our prediction gains relative to M1 show up as negative values (image 33). The more negative we get, the more we have gained over all the frames up to the current point. The steepness of the curve gives us an impression on how much we gained at each frame. We see that starting at around frame 60, our Model M3 shows a significant gain vs. M2 (which is then lost again at a later time after around frame 160).

When examining the event list of “SimpleSequence”, we see that at frame 97 square no.2 is just popping into the scene at the top left, then moving down and to the right, until reaching the bottom right corner of the screen, then popping out again in frame 163. Next, we will take a closer look at how the subjects gaze moved during this period.
In the close-up view of the first 250 frames of “SimpleSequence” (image 34), we see the difference in the curves of M2 and M3 in more detail. When aligning with the gaze shifts (image 35), we understand that this prediction gain is not introduced through one (or a few) saccades, but through tracking (pursuit). This can be explained when looking at the event description: usually, the subject’s gaze is related to saliency only once in a while (see chapter 4.4), not allowing for learning high-value coefficients in the saliency matrix $B$. However, at this particular time, the only “action” is the single square no. 2 popping up at
the top left and then moving to the bottom right of our viewing screen. As this happens, the subject's gaze is obviously attracted by the only available salient region and is therefore moving along the square's path. Since the square's movement is uniform, the motion of the subject's gaze point relative to the candidate point(s) is very low, resulting in an almost static projection of the candidate points to the gaze point. This permits for the saliency matrix $B$ to learn this static projection, with the same results as if we had increased the learning rate for the temporal matrix $A$; the linear motion of the square is predicted with increased efficiency.

So, both the temporal prediction (as found in M2 and M3) as well as the saliency-based prediction (found only in M3) will predict the square's path rather well; in M3, the two prediction gains sum up and together they form a distinct advantage over the single, purely temporal prediction of M2. This can be seen as evidence for the functionality of our algorithm M3 with correct saliency data as an input. On the other hand, later in the sequence, there are more than one salient regions at a time; With the minimum model that we applied ($M = 1$), this sometimes leads to the picking of a wrong (or poorly chosen) candidate point for $S_t$. Obviously, such input data will not lead to prediction gain; it can even harm our overall prediction accuracy, as can be seen most easily at around frame 475 and the following (image 33): the candidate points chosen at this time did not correspond well to those that met the subject's actual attention.

To better illustrate this point we display the individual errors of M2 and M3 and the direct subtraction (images 36,37,38,39).
We shall now take a brief look at the values for the saliency matrix and the temporal matrix in our test runs.

At the starting frame, both matrices were initialized with 0 values, which, had there been no learning effect, would have led to results identical to model M1.

Since M2 is designed to learn short-term trends, its coefficients are not stable for the duration of the sequence; yet there is a certain average trend that can be recognized in the matrix’ values at the end of each test run. Typically, the matrices for our test runs looked like this with “SimpleSequence”:

<table>
<thead>
<tr>
<th>Matrix A for model M2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>L=2</strong></td>
</tr>
<tr>
<td>[ A = \begin{pmatrix} 0.188 \ 0.136 \end{pmatrix} ]</td>
</tr>
<tr>
<td><strong>L=4</strong></td>
</tr>
<tr>
<td>[ A = \begin{pmatrix} 0.052 \ 0.049 \ 0.047 \ 0.046 \end{pmatrix} ]</td>
</tr>
</tbody>
</table>

In both examples, it looks as if the matrices value the more recent gaze shifts more than those further back in time. Yet, since the gaze shifts in \( P_{t-1} \) are all expressed relative to the last known gaze position, the shift lengths that are input into M2 are effectively larger the older they are; so the matrix \( A \) really values the more recent input values equal to or even less than the older ones. It should be mentioned that prediction gains were better with \( L=2 \) than with \( L=4 \) (which is why we chose \( L=2 \) for the previous plots).
With model M3, the behavior of matrix A is quite similar to its behavior in Model M2; the values are just slightly lower, which is probably due to the distribution of the error between the matrices A and B. Matrix B is very similar for both runs and contains relatively small coefficients (see chapter 4.4). Unlike matrix A, the values in matrix B are associated not with increasing temporal delay, but with decreasing image resolution corresponding to our image pyramid.
4.5.2 Sequence no. 2 “AldiKreuzung”

As mentioned in the “Problems and Pitfalls” chapter, the effective gain of M3 vs. M2 is affected by the much more complex distribution of saliency over the scene. While in “SimpleSequence” there were only 1 or 2 salient objects (though each with a small amount of features), in “AldiKreuzung” there are now much more. At some times, there are as much as 20 candidate locations at the same time on the most detailed pyramid level.

![Image 40: Error histogram of M1 with "AldiKreuzung"](image)

![Image 41: Error histogram of M2 with "AldiKreuzung"](image)

![Image 42: Error histogram of M3 with "AldiKreuzung"](image)

The histograms for “AldiKreuzung” (images 40,41,42) show a similar effect as those for “SimpleSequence”, but in a weaker expression, likely caused by the more complex nature of the scene, which can be seen in image 43, page 40.
Compared to the error distances of M1 with “SimpleSequence”, the subject is exhibiting a much more complex gaze pattern with “AldiKreuzung”, containing many more saccades in particular. The effective gain of M3 and M2 relative to M1 is relatively small (see image 44) compared to the results for “SimpleSequence”; this is very likely due to the complex structure of the saliency, causing the chosen candidate location to be of a low average significance for the subject’s actual process of gaze direction. It can be seen that M3 is sometimes loosing, sometimes gaining against M2, with a relatively long period of parallel development (image 44).
The matrix values at the end of the test run for M2 with “AldiKreuzung” is shown in the following table:

<table>
<thead>
<tr>
<th>Matrix A for model M2</th>
</tr>
</thead>
<tbody>
<tr>
<td>L=2</td>
</tr>
<tr>
<td>[ A = \begin{pmatrix} 0.104 \ 0.047 \end{pmatrix} ]</td>
</tr>
</tbody>
</table>

Again, the gaze shifts in \( P_{t-1} \) are expressed relative to the last known gaze position, so the matrix \( A \) weighs the most recent gaze position just slightly more than the one before that.

<table>
<thead>
<tr>
<th>Matrices A and B for model M3</th>
</tr>
</thead>
<tbody>
<tr>
<td>L=2, M=1</td>
</tr>
<tr>
<td>[ A = \begin{pmatrix} 0.104 \ 0.063 \end{pmatrix} ]</td>
</tr>
<tr>
<td>[ B = \begin{pmatrix} -0.0000 \ 0.0030 \ -0.0017 \ 0.0007 \end{pmatrix} ]</td>
</tr>
</tbody>
</table>

For M3 the weight of the older gaze position increases, meaning that the temporal part of the prediction increases its inertia relative to M2. This appears plausible, as the saliency information could be used to predict the more sudden changes, whereas the temporal information could be used to predict the still short-term, yet not so sudden trends; Still, the coefficients of the saliency matrix are very small. Why there are positive and negative values remains to be examined.
5 Discussion and Outlook

The primary goal of our current work is to better understand human eye movements when watching natural, dynamic scenes. In contrast to many other models, our goal is to perform on-line predictions based on the locations attended in the past and the computed salient features of the current (and past) images. Our model shows a potential for analyzing the strategies humans employ when designing their scan paths as well as the evaluation of the saliency of different image features; we started with the most simple assumptions about the perception of saliency and the decision process based thereon. Learning a linear mapping appears to be the most straightforward approach when trying to predict the attended location with a given visual input. Of course, human vision is a much more complex system than a mere linear mapping. To accommodate this, our model has the potential to be extended in several ways. Currently, our prediction is based on finding the optimum weights to scale earlier displacement vectors with a scalar multiplication. A more general model could be designed to also rotate these vectors according to some modified learning algorithm. Also, significantly more candidate locations, both from the present and past images could be used, and the history of attended locations could be extended. As of now, the prediction of saliency-based gaze shifts and temporal trends are both welded into a combined linear mapping. Non-linear mapping (and learning) techniques, e.g. neural networks, will be needed to separate the prediction of temporal trends and saliency-based gaze shifts.

Many attempts have been made to predict the scan path of an observer viewing a static image. We believe that maybe the human visual system may not be optimized for viewing static images, and so random and top-down influences become more dominant when viewing such scenes. We believe that with a dynamic visual input the eye movements are more natural and easier to predict as the bottom-up influences will be more dominant when the input prevents a certain kind of idleness in the visual system. Still, different observers may have very different scan paths and, therefore, see different things with the same visual input. For this reason, we designed our model with the capability to adapt to a particular subject (and a particular visual data stream), and for the same reason we intend to not only observe the scan path, but to also modify it with the goal of improving visual communication systems and vision-based man-machine-interaction in general. We believe that our model can be a first step in doing that.

Summarizing, our model has shown a small, but promising prediction gain with given, suitable input data. Further research is in order to extend and improve our model.
6 Mathematical addendum

The structure tensor

We consider image sequences defined by their respective intensity function \( f(x,y,t) \) and construct the following matrix from the first-order derivatives \( f_x, f_y \) and \( f_t \) of \( f \):

\[
D(x,y,t) = \begin{pmatrix} f_x & f_y & f_t \\ f_y & f_y & f_t \\ f_t & f_t & f_t \end{pmatrix}
\]

The matrix \( J \) is then constructed by convoluting \( D \) with a smoothing kernel \( w \) that is useful for suppressing noise in our image sequences:

\[
J = w \ast D = w \ast \begin{pmatrix} f_x^2 & f_x f_y & f_x f_t \\ f_x f_y & f_y^2 & f_y f_t \\ f_x f_t & f_y f_t & f_t^2 \end{pmatrix}
\]

Due to its construction, \( J \) has been called “the structure tensor” [22].
Derivation of the learning rule for M2 and M3

M2 in more detail:

\[
X_{t}^{\text{pred}} = X_{t-1}^{\text{ref}} + A_{t-1} \times P_{t-1}
\]

\[
\begin{bmatrix} x_{t}^{\text{pred}}, y_{t}^{\text{pred}} \end{bmatrix} = \begin{bmatrix} x_{t-1}^{\text{ref}}, y_{t-1}^{\text{ref}} \end{bmatrix} + \begin{bmatrix} A_{1}, A_{2}, \ldots, A_{N} \end{bmatrix} \times \begin{bmatrix} x_{t-1}^{\text{ref}} - x_{t-2}^{\text{ref}}, y_{t-1}^{\text{ref}} - y_{t-2}^{\text{ref}} \\ x_{t-1}^{\text{ref}} - x_{t-3}^{\text{ref}}, y_{t-1}^{\text{ref}} - y_{t-3}^{\text{ref}} \\ \vdots \\ x_{t-1}^{\text{ref}} - x_{t-N}^{\text{ref}}, y_{t-1}^{\text{ref}} - y_{t-N}^{\text{ref}} \end{bmatrix}
\]

Note that the values in \( P_{t-1} \) are expressed relative to the last known reference position.

We are intending to minimize the cumulative quadratic error for M2:

\[
E_{M2} = \frac{1}{T-1} \sum_{t=2}^{T-1} \left( X_{t}^{\text{ref}} - \left( X_{t-1}^{\text{ref}} + A_{t-1} \times P_{t-1} \right) \right)^{2}
\]

The incremental learning rule is \( A_{t} = A_{t-1} + \epsilon \text{ } e_{i} P_{i}^{T} \) or, more applicable here:

\[
A_{t} = A_{t-1} + \alpha \epsilon_{A} e^{T} P_{i}^{T}
\]

where \( \epsilon_{A} \) is the learning rate for matrix \( A \) and \( e = X_{t}^{\text{ref}} - X_{t}^{\text{pred}} = X_{t}^{\text{ref}} - \left( X_{t-1}^{\text{ref}} + A_{t-1} \times P_{t-1} \right) \) the current error.

The direction of the gradient along which we descend is \( e \times P_{t-1}^{T} \).

Finding the appropriate learning rate is done by minimizing the quadratic error:

\[
\min \left( \left( X_{t}^{\text{ref}} - X_{t-1}^{\text{ref}} \right) - \left( A_{t-1} + \epsilon_{A} e^{T} P_{i}^{T} \right) P_{t-1} \right)^{2}
\]

\[
\min \left( X_{t}^{\text{ref}} - X_{t-1}^{\text{ref}} - A P - \epsilon_{A} e^{T} P_{i}^{T} P \right)^{2}
\]

\[
\min \left( e - \epsilon_{A} e^{T} P_{i}^{T} \right)^{2}
\]

\[
\epsilon_{A} \rightarrow \frac{d}{d \epsilon_{A}}
\]

\[
e_{A} \quad e_{A} \rightarrow e_{A}
\]

\[
\epsilon_{A} e_{A} P_{i}^{T} P \rightarrow 0
\]

\[
= \epsilon_{A} e_{A}^{T} P_{i}^{T} P P_{i}^{T} P e_{A}^{T}
\]

\[
= \frac{|e_{A} P_{i}^{T} P e_{A}^{T}|}{|P_{i}^{T} P e_{A}^{T}|^2}
\]

J. Drewes Institute for Neuro- and Bioinformatics, University of Lübeck, 2003 44/49
Derivation of the learning rule for M3

M3 in more detail

\[ X_{t}^{\text{pred}} = X_{t-1}^{\text{ref}} + A_{t-1} \times P_{t-1} + B_{t-1} \times S_{t} \]

\[ = X_{t-1}^{\text{ref}} + \left[ A_{1, \ldots, A_{N}} \times \begin{bmatrix} X_{t-1}^{\text{ref}} - X_{t-2}^{\text{ref}} \\ \vdots \\ X_{t-1}^{\text{ref}} - X_{t-N}^{\text{ref}} \end{bmatrix} \right] + \left[ B_{1, \ldots, B_{N \times A}} \times \begin{bmatrix} X_{t-1}^{\text{ref}} - X_{t,1}^{\text{C}} \\ \vdots \\ X_{t-1}^{\text{ref}} - X_{t-N,1}^{\text{C}} \end{bmatrix} \right] \]

Notice here as well that the values in \( P_{t} \) and \( S_{t} \) are expressed relative to the last known reference position.

In analogy to the proceedings with M2 we are minimizing quadratic errors, for matrix \( A \) as above and for matrix \( B \) as in:

\[
\min \left[ \left( X_{t}^{\text{ref}} - X_{t-1}^{\text{ref}} \right) - \left( B_{t-1} + \epsilon_{B} y_{B} S_{t}^{T} S_{t} \right) \right]^{2}
\]

\[
\min \left[ \left( X_{t}^{\text{ref}} - X_{t-1}^{\text{ref}} \right) - B S - \epsilon_{B} y_{B} S^{T} S \right]^{2}
\]

\[
\min \left[ e_{B} - \epsilon_{B} y_{B} S^{T} S \right]^{2}
\]

\[
e_{B} \rightarrow \frac{d}{d \epsilon_{B}} \quad e_{B} y_{B} \rightarrow e_{B}
\]

\[
\begin{bmatrix} e_{B} - \epsilon_{B} y_{B} S^{T} S \end{bmatrix} \begin{bmatrix} e_{B} y_{B} S^{T} S \end{bmatrix}^{T} = \begin{bmatrix} 0 \end{bmatrix}
\]

\[
e_{B} y_{B} S^{T} S \begin{bmatrix} e_{B} y_{B} S^{T} S \end{bmatrix}^{T} = \frac{e_{B} S^{T} S e_{B}^{T}}{\| S^{T} S e_{B}^{T} \|^{2}}
\]
7 Image Listing

Image 1: Schematic view of the human eye [1] 2
Image 2: Schematic view of the retina [1] 2
Image 3: Distribution of rods vs. cones [1] 3
Image 5: Typical scan path (abbreviated) [2] 4
Image 6: Image perception through foveated vision (simplified) 4
Image 7: Same image but different messages [3] 4
Image 8: Attention - aware driver's assistant [3] 5
Image 10: A box 10
Image 11: Boxes i1D features (symbolic) 10
Image 12: Boxes i2D features (symbolic) 10
Image 13: Box popping up, according S values, resulting candidate points 14
Image 14: The learning mechanism 19
Image 15: Algorithm summary 20
Image 16: Sample frame no. 300 from "SimpleSequence" 23
Image 17: S-valued saliency for frame no. 300 (SimpleSequence) 23
Image 18: K-valued saliency for frame no. 300 (SimpleSequence) 23
Image 19: Sample frame no. 330 from "AldiKreuzung" 25
Image 20: S-valued saliency for frame no. 330 (AldiKreuzung) 25
Image 21: K-valued saliency for frame no. 330 (AldiKreuzung) 25
Image 22: Pupil and corneal reflexes, tracked by iViewX 26
Image 23: System setup with test subject 26
Image 24: Example for composited output image ("SimpleSequence") 27
Image 25: Subject's gaze shifts for "SimpleSequence" 30
Image 26: Prediction errors of M2 for "SimpleSequence" 31
Image 27: Difference of M2 and M1 for "SimpleSequence" 31
Image 28: Prediction errors of M3 for "SimpleSequence" 32
Image 29: Difference of M3 and M1 for "SimpleSequence" 32
Image 30: Error histogram for M1 with "SimpleSequence" 33
Image 31: Error histogram for M2 with "SimpleSequence" 33
Image 32: Error histogram for M3 with "SimpleSequence" 33
Image 33: Prediction gain of M2 vs. M3 relative to M1 with "SimpleSequence" 34
Image 34: Prediction gain of M2 vs. M3 relative to M1 with "SimpleSequence", 0-250 35
Image 35: Subject's gaze shifts with "SimpleSequence", frames 0-250 35
Image 36: Prediction errors of M2 with "SimpleSequence", frames 0-250 36
Image 37: Prediction errors of M3 with "SimpleSequence", frames 0-250 36
Image 38: Difference of M2 and M3 with "SimpleSequence", frames 0-250 37
Image 39: Cumulated difference of M2 and M3 with "SimpleSequence", frames 0-250 37
Image 40: Error histogram of M1 with "AldiKreuzung" 39
Image 41: Error histogram of M2 with "AldiKreuzung" 39
Image 42: Error histogram of M3 with "AldiKreuzung" 39
Image 43: Subject's gaze shifts with "AldiKreuzung" 40
Image 44: Cumulated gain of M2 vs. M3 relative to M1 with "AldiKreuzung"
8 Bibliography

2: Image courtesy of Guiseppe Boccignone, Università degli Studi di Salerno, Fisciano, Italy
3: E. Barth, T. Martinetz, Information technology for active perception, 8th Annual German-American Beckman Frontiers of Science Symposium, 2002
5: B. M. McKay, Behind the eye, 1991
7: O. Regan, A. Noe, A sensorimotor account of vision and visual consciousness, 2001
12: L. Itti, Models of Bottom-Up and Top-Down Visual Attention, 2000
15: R. Milanese, S. Gil, T. Pun, Attentive mechanisms for dynamic and static scene analysis, Optical Engineering, 34, 2420-2434, 1995
16: G. Boccignone, A. Marcelli, G. Somma, Analysis of dynamic scenes based on visual attention, 2002
17: E. M. Helveston, The Strabismus Minute, 2000
18: C. Mota, E. Barth, On the uniqueness of curvature features, 2000
22: Barth, E., The minors of the structure tensor, 2000
9 Acknowledgements

I would like to extend my deepest gratitude to all the people who helped me to create this thesis.
Some of the source codes in my project are based on original routines for a motion estimator programmed by Martin Haker. The hard-synced viewing application, without which we would not have gotten a good synchronization between our sequence and the measured gaze data, was written by Manuel Wille. Together with him I experienced the joys of reliable, high-speed parallel port output under windows 98.
Special thanks go to Bogdan Anastasiu, who had to endure the fate of sitting opposed to me in the same office. Being my favorite test subject for long weeks (or months, rather) of testing, he probably knows our video sequences better than myself now.
I would also like to mention Amir Madany for helping me out with lots of “trivial” syntax problems both in Matlab and C. In this context I would also like to thank Jan Kim for his many polite ways of saying “rtfm”; as that is just what one should do first.
Cicero C. Mota supported me by providing mathematical additions and double-checking my own creations.
Dr. Erhardt Barth, my mentor, deserves my deepest gratitude for guiding me and devoting a lot of his time (sometimes more than he actually had) into supporting me, even when his phone was keeping him quite busy.
Aside from taking personal interest in the progress of my work and participating through his contributions, Professor Thomas Martinetz, the head of the INB also had the faith to send me, as an undergraduate student, off to a convention to present this project (as a poster).

It has been a magnificent experience to work with the people at the INB, and I will always remember this time with feelings of joy and pleasure.