An eye-tracking based approach to gaze prediction using low-level features

Erik Johannesson

Master's Thesis
Spring 2005

Department of Cognitive Science
Lund University
Sweden

Abstract

In this master's thesis, an attempt is made to automatically predict where people will look when watching video sequences. An application in the form of foveation for video compression is discussed. A relatively simple prediction model is built based on eye-tracking data from several subjects and low-level features generated from the video frames, using simple image processing algorithms. The prediction model uses a new method to extract differences in feature distributions between frame regions that are watched and those that are not. It is first shown that these differences are significant. The differences are then used to predict which regions will be looked at in a new video sequence. The prediction is evaluated against eye-tracking data for the new video sequence and it is shown that the prediction is significantly better than random. Moreover, the accuracy of the prediction is compared to that of a group of humans predicting another group of humans. This comparison indicates that the proposed model needs improvement. Finally, a discussion follows about the possibilities and problems of the selected approach to gaze prediction.

Acknowledgements

I would like to thank the following persons for helping me in many invaluable ways:

Kenneth Holmqvist, for supervising with patience and helpful guidance. Marcus Nyström for providing input and interesting articles. Finn Lindgren for kindly helping me with statistics. The other master's students in the eye-tracking group have been very helpful as well.

A special thanks I would like to give to everyone who participated in the experiment. I will also not forget those who let me use their computers, for many days and nights, to do the necessary calculations. I'm lucky to have such good friends.

1 Introduction

1.1 Improved video compression by foveation

There has been, and still is, an enormous growth in the use of digital video. New applications, such as video telephony or Internet video streaming, and older ones, such as digital video home recording, continue to increase in popularity. Considering this, the need for efficient compression technologies is undeniable, given that transmission and storaging capabilities are still limited.

Traditional video compression methods assume that every part of a video frame is equally important. Consequently, every part of a frame is equally degraded when the video is compressed. However, the assumption is not true since the human eye is only capable of sharp vision in a limited area. The visual acuity starts to drop rapidly at an angle of about 2° from the line of sight (Duchowski, 2002). This means that a relatively higher level of degradation in the periphery would not matter as much to the viewer as it would in the line of sight. Following this reasoning, it seems rational to use a higher compression rate for information that is of less importance, while providing a better image in the interesting parts of every frame. If this technique, known as *foveation*, were to be used by video compression software, it would be possible to increase the efficiency of compression – and get a better trade-off between quality and bit rate. See for example (Wang & Bovik, 2001) for an introduction to foveation.

The problem with foveation is that it is not known in advance where a person will look on a video frame. There are currently three different solutions to this problem. The traditional solution, called *online foveation,* is to provide the person watching the video with eye-tracking equipment and feed the coordinates to a computer that renders the video differently based on where the subject is looking. However, this is not a plausible solution in many cases since eye-tracking equipment is expensive and complicated to use. Also, this approach does not reduce the requirements in terms of storage and bandwidth since the compression then would have to be made ad hoc. This solution is rather suitable

for applications where the video is rendered in real time, such as flight simulators (see Jacob & Karn, 2003, for a description).

Another approach to solving the problem is to show the video that is about to be foveated to a group of people and measure their eye movements. The collected data can be used as a predictor of where people will look when seeing that video sequence, and to foveate it accordingly. This can be called *offline foveation* has been done successfully (Nyström, Novak & Holmqvist, 2004). In most cases however, this procedure is too costly to be used in practice, which led to the third solution approach.

The third approach is to create an algorithm which in some way tries to predict, for every frame in a video sequence, what areas that are likely to be looked at, without the need for experiments with live subjects. This approach can be referred to as *predicted foveation.* Such an algorithm would indeed provide a cost-efficient way of improved digital video compression. This thesis describes an attempt to build a gaze prediction model, upon which such an algorithm is implemented (hereafter the word *model* will be used to signify both the model and the algorithm). However, as we shall see next, it is not at all obvious that it is possible to build a model capable of delivering accurate predictions of where people will look.

In addition to improved video compression, other applications could also benefit from a gaze prediction model. The model could be useful in applications where it is important to extract the most important regions in an image or a video sequence, such as robotic vision, surveillance, automatic video classification etc. Beside the applications, the (validation of the) model could also generate knowledge about what influences gaze behavior, which is of theoretical interest.

1.2 Gaze prediction – a futile endeavor?

Is it possible to determine in advance where a person will look when watching a video sequence? There are good arguments both for why it could, and why it could not be possible. One could argue that the contents, or chain of events, of the video would to a

high degree control where one looks. If, for example, you are watching a film where someone suddenly enters an empty room and starts to speak, the chances are high that you would look at that person. A good argument comes from eye-tracking research that has shown that faces, especially the areas around the eyes and the mouth, tend to attract a great deal of eye fixations (see for example Henderson et al, 2000; Gullberg & Holmqvist, 1999; or Yarbus, 1967). This kind of knowledge could possibly, together with properties of the video sequence, be used for prediction of where people will look.

On the other hand, it can be argued that humans have free will and prefer to look at different things. The randomness of people's will and interest would thus make it impossible to predict the movement of their eyes. In the early days of eye-tracking research it was found that even just a single person looked at an image in completely different ways depending on a given task. In particular, some areas of the image were more looked at for one task and less for another (Yarbus, 1967) Thus, one could also argue that any attempt of prediction might have to consider the "task" that the person watching the video sequence is performing, whatever that might be.

Some attempts have been made at predicting where people will look on still images. When images are exposed briefly to subjects, some previously proposed algorithms seem to be able to somewhat accurately predict *where* the subjects will look. The prediction of the *order* of fixations however, seems much harder (Privitera & Stark, 1998).

However, contrary to the standpoint that gaze is too random and personal to predict, there is experimental evidence that the differences in people's eye movement when watching video are quite limited. It has been shown that gaze points tend to group in one or few clusters (Dorr et al, 2005). An interesting result is that when eye-tracking data from a group of subjects was used to foveate a video sequence (as described in section 1.1 as the offline foveation approach) and it was shown to a second group of subjects, it was found that their gazes were more clustered compared to the first group. Apparently, foveation steers gaze points into the less compressed area(s) (Nyström, Novak & Holmqvist, 2004). It has also been shown that there are no significant differences in gaze patterns due to age

or gender (Goldstein et al, 2004). The essence of these results is, among other things, that the possible existence of interpersonal "tasks" for video watching is negligible, and that the problems of predicting the fixation order in still images does not carry over to moving images. The latter could be because video watching to a higher extent than still image watching is driven by stimuli, whereas still image watching is more of a top-down process since there is more time to freely explore every image. With all this in mind, gaze prediction in video seems a lot more plausible.

The standpoint that prediction of gaze points is not possible could still be valid in one sense: Given that people have free will, a person could choose to just look at an arbitrary point on the screen without caring about what happens in the video. That kind of behavior is unlikely and seems impossible to predict. To require that is however a bit too strict. One has to remember that prediction does not necessarily mean being able to say exactly how one person will behave. Instead, prediction often refers to specifying what will happen on average. Take this as an example: One could predict that the weather in Lund will be mostly sunny in July, in a majority of years in the future. This would probably make quite a good prediction, although there will of course always be those summers when it rains all the time. When that occurs is much harder to say. Still, the prediction is often right and so it is useful. This reasoning applied to prediction of eye movements says that it will be impossible to say exactly where *someone* will look, but predicting where the majority of a *group of people* will look is still possible.

In conclusion, the requirements for a successful prediction are that people tend to look at the same regions of video frames and that there are identifiable *features* in the video frames that in some way are correlated to the likelihood of different regions being looked at. Logically, the second requirement follows from the first since there could hardly be any other reason for the earlier mentioned tendency of clustering of gaze points, than that there is some feature in the video frame that correlates with gaze amount. What could otherwise attract different persons gaze to the same regions? The interesting question is thus not if there are features that correlate with gaze, but what those features are. Even if they do exist, they could be connected to some high-level cognitive aspects and as a

result be difficult or impossible to quantify. Prediction based on such features could indeed become quite a hard task.

A model such as the one proposed here, assumes that the correlations between gaze behavior and some specific, computable, low-level features are sufficient for prediction purposes. Such correlations have been shown to exist for still images (see Reinagel & Zador, 1999; Privitera & Stark, 2000). Also, correlations have been found for video (Itti 2005). Nevertheless, the assumption of correlation for video will constitute a part of the hypotheses of this thesis, and will be tested for the specific features that are used in the model described here. The main reason for this test, even though correlations already have been found, is that a different approach will be tried in how the features are used.

The rest of the introduction section contains a quick presentation of the proposed prediction model, a comparison with another gaze prediction model, and the hypotheses of this thesis. The second section contains a description of the data material that has been used, and the experimental conditions under which it was collected. Section 3 is focused on presenting the features that are used in the model. In section 4, the data material is analyzed with regards to the used features, and it is investigated if the features correlate with gaze points. Section 5 is devoted to the prediction of gaze points for video and validation of this prediction. The results of the data analysis and the prediction are finally discussed in section 6.

1.3 The gaze prediction model

The model described in this thesis (hereafter referred to as *our model*) will here be shortly described. A more careful treatment is given in sections 3, 4 and 5. Our model is based on two things: eye-tracking data that was recorded for several subjects while they were watching video sequences, and a set of low-level features. Practically no knowledge of the human visual system (i.e. the parts of the brain that are involved in visual processing and attention) is assumed.

A low-level feature is, in one sense[1], a quantitative measure generated by an image processing algorithm using one or more frames of the video sequence. For example, a simple low-level feature is the luminance (brightness) of different points in a frame. An image processing algorithm that calculates the luminance feature would take an image (a video frame) as input, and return a new image whose pixel values are proportional to the luminance of the pixels in the original image. The result would look like a gray-scale version of the original image. Another example of a low-level feature is what is returned by an image processing algorithm that emphasizes edges (an *edge detector*). The algorithm transforms an image into a new image where the pixel values are related to the difference in intensity between adjacent pixels. An example of such a transformation, for a gray-scale image, can be seen in figure 1. In the resulting image, edges are given high values and the areas between them are given lower values. All in all, our model uses different variants of four basic features: Intensity, edges, motion and contrast.

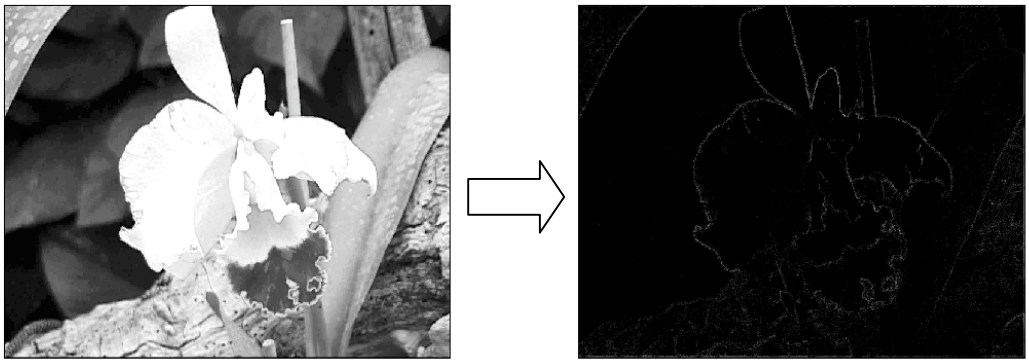


Figure 1: Example of edge detection

[1] Two different interpretations are given to the term *feature*. The first interpretation, as discussed in the previous section, refers to visual properties of objects at any level. This is a modification of the feature term used in the visual search literature, where *basic features* are stimulus attributes that supports efficient search and effortless segmentation of objects (Wolfe, 1998). The other interpretation is that a feature is a computable measure of an image (or several consecutive images). There is often a clear correspondence between the two interpretations (as for the color feature), but not necessarily (as for the 'edge feature'). When the term is used, the context can be taken as a guide to which interpretation is appropriate. Although hereafter, the word will mainly be used to signify computable meauses of images.

The model workflow can be seen in figure 2. Our model works in two main phases. First there is the *analysis* phase (which is similar to what is often referred to as training in machine learning literature). This phase takes eye-tracking data and the corresponding video sequence(s) as input. It consists of calculating measures of the different features for the video that has been shown to the subjects. The feature measures are calculated for the regions of the frames that have been looked at, and for the frames in total. For reasons that will be explained later, the contrast feature measure is not calculated for the whole frames, but in random regions of the different frames. These feature measures are stored and analyzed to construct statistical distributions, here called *feature density maps*, which is the output of the analysis phase. These maps describe how certain values of a feature correlates with gaze, and are in structure similar to probability functions. The information contained in the maps is an aggregate of the recorded eye-tracking data, with regard to the features.

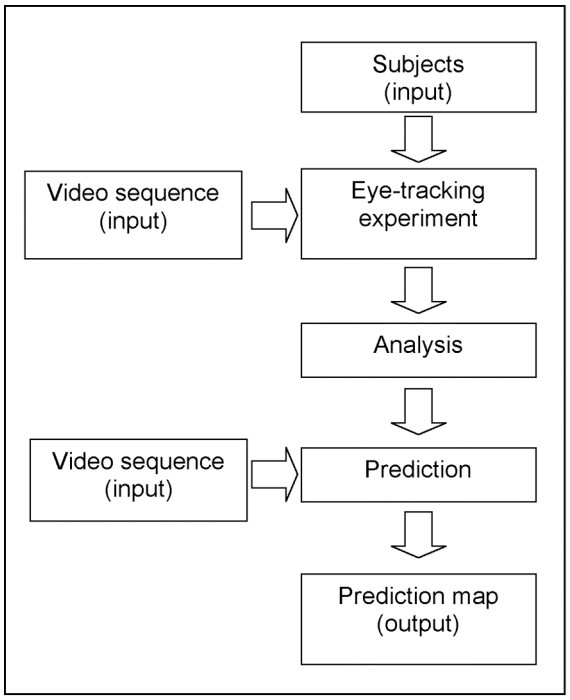


Figure 2: The model workflow

Since our model works through statistical analysis, it does not assume any knowledge about how the real human visual system functions. An exception however, lies in the selection of features used in the model. These were chosen with consideration of previous research about what attracts gaze and visual attention (Wolfe, 98; Reinagel-Zador 1999; Itti 2004; Privitera & Stark, 1998).

The other step is the *prediction* phase. It takes a set of feature density maps and a video sequence (which has not been used in the analysis phase) as input. The prediction is carried out by calculation of feature measures on the frames in the video. The different

feature density maps are then used together with these measures to decide the likelihood of a gaze in different regions of the video frames. The general idea is that the likelihood is calculated based on what value the feature measures in the actual region correspond to in the feature density maps. The output is finally presented as two-dimensional probability[2] maps, here called *prediction maps*[3], which are supposed to describe the likelihood of gaze in different points of the frame. The prediction maps can then be validated against eye-tracking data (this is done in section 5). If the prediction is to be used for foveation, the video codec is supposed to alter the compression rate based on the prediction maps.

1.4 Other attempts at prediction, similarities and differences

There has been some research on the prediction of gaze points and related concepts[4] (see for example Osberger & Maeder, 1998; Privitera & Stark, 2000; Itti & Koch, 2000). Also (Ericsson & Pehrsson, 2005) is relevant in that the methodology is in part similar to what is tried here.

Only recently, research has turned to the interesting problem of predicting gaze points in video. The most well-known attempt is briefly described in the following subsection.

---

[2] Strictly, they are estimations of probability functions which are assumed to exist for the purpose of this thesis.

[3] Maps with the same function are sometimes called saliency maps, to point out that they estimate the relative saliency of different regions on the screen (Itti, 2004).

[4] Some prefer using the concept *region of interest* (ROI) to denote regions where people look. A gaze point is here assumed to be located in a ROI. Others prefer to predict *fixations*. Essentially this is the same thing as predicting gaze points, although only points where the eye fixates for a certain time period are counted (i.e. gaze points during saccades are not counted). In the actual prediction, it makes no difference to predict gaze point or fixations. The reasons that this master's thesis uses the more general concept of gaze points is that it would be impractical to filter out fixations from the data material.

1.4.1 The iLab model

A model that has gotten a lot of attention lately has been created by iLab at University of Southern California[5], led by Laurent Itti (the model will hereafter be referred to as the iLab model). The most important aspects of the iLab model will be described here in short, but see (Itti, 2004) for details.

The iLab model is originally based on a visual search model for still images (Itti & Koch, 2000) with added temporal aspects. Both models are based on neurobiological knowledge and assumptions about the visual system. The iLab model calculates feature maps for twelve different low-level features at nine spatial scales. It then calculates a center-surround difference for every feature map, at different pairs of scales. This yields six maps for every feature, a total of 72 feature maps. These maps undergo a form of internal competition, where regions with high feature values compete with each other to gain high values in the final maps. The competition is then repeated across the spatial scales. The result indicates those regions that stand out from their surroundings. This is an attempt to model the pop-out effect in visual search (described in Wolfe, 1998). The maps are finally added to create what is called a saliency map, which is used to guide the prediction (some normalization, scaling and temporal smoothing is also performed). Optionally, it is also possible to a priori define a discrete number of virtual foveas (simulated gazes) that are driven by the saliency maps and have internal dynamics. This option is probably a heritage from the visual search model, and could perhaps be useful for applications such as video compression or robot vision.

The greatest difference between the iLab model and our model, is the starting point. Instead of trying to approximate neurobiological phenomena, our model is based on a statistical analysis of gaze points and their surroundings in the video frames. Our model is simpler, but the structure is harder to motivate from a theoretical perspective. But even though there is no attempt here to construct a model for how the human visual system actually works, the model is implicitly a function of the neurobiological reality. This is

[5] See http://ilab.usc.edu

because the eye-tracking data, that the model is built upon, contains what real human visual systems (of test subjects) have produced.

The iLab model, on the other hand, is built from theorizing – not with regards to the types of features involved (they have all been proved to correlate with eye-tracking data), but there is no experimental data involved in the actual modeling. This is not necessarily bad, since it makes the model context-free and independent of what data material is available, but at the same time it seems awkward to build a model without using data to control any of the parameters. One could assume that the creators of the iLab model have very good knowledge about the human visual system, or maybe suspect that the model is the result of an advanced trial-and-error scheme, or perhaps a combination of the two. It does leave room for questions such as whether the relative weighting of the features is optimal. All features are currently treated as equally important which is not in agreement with the finding that some of the features have higher correlation with eye-tracking data, as found in (Itti, 2005). If a particular feature correlates more with gaze than others, why should not a model of the human visual system take that into account? On the other hand, there are also drawbacks with only using experimental data to build the model, as done here. It becomes very sensitive to the data material used. If the data material is non-representative, the predictions will probably not be very good for most video sequences.

The two models are clearly built with different purposes. The main goal for iLab seems to be biological plausibility, which is not considered here. The primary matter of interest in this thesis is the predictive ability of the model. Our model does therefore not include the option of having virtual foveas. Instead, the prediction maps in themselves are supposed to be used for foveation of video. Another aspect, which is important for applications, is that our model is simpler and computationally faster[6]. For example, it does not use different spatial scales, and there are no competitive networks involved.

[6] This is, of course, given that both algorithms and their implementations are optimized to the same degree.

But perhaps one of the most important differences is the extent to which the two models have been validated: Our model has been built and validated on eye-tracking data from groups of 14 and 26 subjects. The iLab model has been validated on a set of video sequences where the number of subjects varies between as few as four and six (average 4.7). Even if the results of the prediction were significantly better than random, the ecological validity of the results are disputable with so few subjects. On a further account, it is questionable how naive Itti's subjects really were. The subjects were given explicit instructions to look at the main events in the video sequences (Itti, 2004).

1.5 Hypotheses

The goal of this thesis is to produce a model that uses computable low-level features together with eye-tracking data to produce gaze predictions, for video, of the best possible quality. As discussed in the introduction, a necessity for prediction is that the features used by the model correlate with the amount of gaze a region receives. Even though this has been previously shown for a set of low-level features, the assumption deserves to be tested yet again. A good reason for this is the considerable amount of data available (see section 2), another is that a different approach has been used for the feature analysis compared to previous studies (such as Reinagel & Zador, 1999; Privitera & Stark, 2000; Itti 2004).

Analysis Hypothesis: *There are significant differences in low-level features between regions that are looked at and regions that are not looked at.*

Even if there are differences, this does not guarantee that they are sufficient for prediction purposes. A minimum requirement for prediction is that it is better than random guessing.

Weak Prediction Hypothesis: *Using only low-level features, it is possible to predict which regions will be looked at. The prediction is significantly better than what could be achieved by random.*

As discussed in section 1.1, it is impossible to exactly predict where a person will look, so to require that accuracy would be unfair. However, since people's gazes tend to cluster, the best possible predictor is probably another group humans. To expect a model to have the same accuracy of predictions as a group of humans is perhaps far-fetched, but in an optimistic spirit, a comparison is nevertheless to be performed.

Strong Prediction Hypothesis: *Compared to a group of humans predicting another group of humans, prediction based on low-level features is not significantly worse.*

To quantify the extent to which these hypotheses are met will be much easier given a more exact description of the analysis and prediction phases. The hypotheses will therefore be operationalized in the subsequent sections, just before they are evaluated with regards to the results.

2 Data material

The data material used in this thesis consists of video sequences and eye-tracking data recorded for these sequences. Some data was available in prior, and some was gathered through an experiment.

2.1 Data from a previous study

Information about video sequence A can be found in table 1. The eye-tracking data was recorded at 50 Hz, using 14 subjects. The video sequence and the eye-tracking data originate from (Nyström, Novak & Holmqvist, 2004) (the video sequence used here is the non-foveated version).

Video sequence A	
Length	3 m, 35 s
Frame rate	25 Hz
#Frames	5384
Resolution	720×576
Image size	720×576
Colors	Thousands
Sound	44100 Hz Stereo
Content	Seven different types of natural scenes

Table 1: Video sequence A

2.2 Experiment

To collect additional data, an experiment was conducted. A new video sequence was

prepared (see table 2). It consists of two different parts, where the second follows directly after the other with a mere interruption of four black frames. The image size was smaller than the resolution, meaning that there were black borders around the video sequence.

The experiment took place in Humanistlaboratoriet, in the premises of Språk- och Litteraturcentrum (SOL-centrum) at Lund University, April 28th and 29th, 2005.

2.2.1 Subjects

30 subjects (18 male, 12 female) were recruited. Most, but not all, were students at Lund University. Their ages were (by estimation) between 20 and 55, with a median around 25. None of the subjects knew in advance about the purpose of the experiment. When asked after the experiment what they believed was the purpose, most responded that it was to see what they had looked at, but they could not say why. A few of them thought that the purpose was to determine what would attract their attention (motion was mentioned by most of these). However, all the subjects were quite vague and could not elaborate. No one mentioned anything about (low-level) features, statistical analysis, prediction, or anything related. Many subjects did however show interest in the film (Boondock Saints), saying that they wanted to see the rest of it as well (only five had seen the film before). The subjects claimed that they felt comfortable with the experimental situation and that they had watched the video as they would have done in a normal setting. Visual inspection of the recorded data does not indicate otherwise (i.e. there were no subjects that did not seem to follow the major events in most parts of the video sequence).

Video sequence B	
Length	Part 1: 4 min, 41 s Part 2: 15 min
Frame rate	25 Hz
#Frames	Part 1: 7026 Part 2: 22497
Resolution	768×576
Image size	Part 1: 352×288 Part 2: 759×329
Colors	Thousands
Sound	Part 1: None Part 2: 44100 Hz Stereo
Content	Part 1: Standard clips used in video compression research Part 2: The beginning of the movie 'Boondock Saints'

Table 2: Video sequence B

2.2.2 Procedure

The subjects were received one at a time. They were seated in a chair approximately 75 centimeters from a Dell 17" tft screen, which was used to display the video sequence. The screen had a resolution of 800x600 and was connected to an Apple Powerbook G4 1GHz in another room that used QuickTime to display the video (in 'normal' size, with the rest of the screen black). An SMI iView X remote camera was placed on the table directly under the monitor and used as an eye-tracking device. An SMI iView infrared corneal reflex pupil system was attached to the sides of the monitor. It was used to emit infrared light that was reflected by the subjects' eyes into the camera. Nothing was attached to the subjects, so they could freely move their heads although they were instructed not to make any quick movements since the camera can only follow slow movements.

The subjects were told that they could abort the experiment at any time, without having to motivate why. They were also informed that they have the right to contact the experiment leader (the author) after the experiment in order to have their data deleted.

The eye-tracking device was calibrated using 9 dots located in a grid, displayed one at a time, on an area of the same size as the video sequence. The subjects looked at each of them while an eye-tracking expert handled the calibration program from the other room. After the calibration, the subjects were told that they should look at the video in the same way they would usually do. They were told that they first were going to see a red dot (see section 2.3 for an explanation) that makes a circular motion across the screen and then a number of short clips with no sound, followed by 15 minutes from the beginning of a movie.

The subjects were given the possibility of asking questions (no questions about the purpose of the experiment were answered). They were finally told to look at the red dot as long as it was on screen. The light was then switched off in the room, and they were left alone as the video sequence began to play.

The eye-tracking device filmed one of the subject's eyes while the video sequence played. A computer then took the images of the eye in the infrared wavelength band, and calculated the relative angle at which the eye was directed towards the screen, using the images of the pupil and the corneal reflex. Using the calculated angle, the on-screen coordinates of the gaze where calculated. Measurements were given at 50 Hz (every 20 ms). The measurement error was estimated to be less than one degree (which equals approximately 30 pixels on the screen).

During the experiment, the eye-tracking expert monitored the subjects and the eye-tracking device from the other room. If the subjects' head movements made the camera loose track of the eyes, this was corrected for within a few seconds.

When the video sequence had reached the end, the experiment leader entered the room and the light was turned on. The subjects were asked for their general impression of the experiment and if they had seen the movie before. They were also asked about their beliefs about the purpose of the experiment. Nothing was however revealed until after the final subject had left, when an explanatory e-mail was sent out to the subjects. Before leaving, the subjects signed a paper saying they permit usage of the recorded data material for academic purposes. Every subject also received a small gift in the form of two lottery tickets.

2.2.3 Resulting data

The result of the experiment was 30 files with on-screen eye-tracking coordinates at 50 Hz (two measurements per frame). No difference is made in the data if the measured coordinates are due to fixations, saccades, smooth pursuit, blinks, or anything else. Measurements recorded when the subject looked outside the screen or closed his/her eyes, or when the camera lost track of the eye, were removed.

Of the 30 subjects, four (two male, two females) was later removed due to different reasons, such as calibration errors. For one subject (female), only 13,5 minutes of data

was later used, due to a sudden error in the video playback 14 minutes into the experiment.

2.3 Post-experimental calibration procedures

In addition to the spatial calibration performed prior to the data recording, a temporal calibration has to be performed as well in order to match the eye-tracking data with the frames of the video sequence. This was done in the same way for both of the two available data sets.

The purpose of the red dot (included in the beginning of both video sequences) is to support the temporal calibration. The calibration is performed by calculation of the center of the red dot for each frame (the diameter of the red dot is 55 pixels). For each frame, the distance between the gaze point and the center of the red dot is then calculated for different temporal offsets in the data. This means that the data is moved back and forth in time while the distance to the red dot is measured for a particular frame. This is done for all the frames where the red dot is visible (the rest of the screen is black in those frames). Of all the temporal offsets that generated the smallest distance, the median is selected as the temporal offset to be used.

After the temporal calibration, the first 300 frames were discarded. Only 161 of these were used for calibration, but removal of the first few seconds reduces any transient effects that may be present in the beginning of the experiment.

A criterion for a subject to be used was that there were at least 20 consecutive frames (40 consecutive measurements where the distance between the gaze point and the center of the red dot was less than 70 pixels. Other than the subjects already mentioned as discarded, all the subjects passed this criterion.

In video sequence B, the image area is smaller than the video resolution. The data coordinates were therefore transformed with a spatial offset, and the video was cropped so that only the image area remained. The main reason to do this is that the feature

calculations would otherwise have been biased due to a big portion of the screen practically never being looked at (e.g. the color black would be attributed an artefactual low correlation with gaze). The cropping also increased the speed of the calculations that followed.

3 Features and images

3.1 Computer representations of images

All the calculations and data handling in this thesis were performed in Matlab. Some specific details that follow might differ in other environments.

Using QuickTime, the video sequences A and B were converted to sequences of tiff images. These were then imported to Matlab for further processing. Color images are represented in Matlab by three-dimensional matrices of size H×W×3, where H is the image height and W is the image width. The values in the matrix are floating point numbers[7] in the interval [0,1] that describe the intensity of one of the three color channels in the pixel that corresponds to the location in the matrix. If the image is in RGB format (as it is when it is imported from a tiff file), then the first layer of the matrix contains the red color channel, the second the green and the third the blue. A completely blue image would thus be represented by zeros in the first two layers and ones in the third layer.

To be able to handle color information in images separately from luminance information, the RGB images need to be converted into another color space. We will here use the YCbCr color space. In this color space the first layer in an image matrix, the Y channel, represents the luminance[8], and the other two layers, the Cb and the Cr channels,

[7] It is also possible to use integer representations

[8] Actually it represents the *luma* which is not exactly the same thing. The term luminance will be used continuously since the difference is not relevant here, but see http://en.wikipedia.org/wiki/Luminance (2005-06-06) for details.

represents chrominance (i.e. color content). Descriptions of the YCbCr color space are readily available[9].

In Matlab there is a function[10] that transforms an image from RGB to YCbCr, using the conversion formula:

$$Y = \left(65.481 \cdot R + 128.553 \cdot G + 24.996 \cdot B + 16\right)/255$$
$$Cb = \left(-37.797 \cdot R - 74.203 \cdot G + 112 \cdot B + 128\right)/255$$
$$Cr = \left(112 \cdot R - 93.786 \cdot G - 18.214 \cdot B + 128\right)/255$$

As can be seen in the first equation, Y only takes values in the interval [0.063, 0.922]. Figure 3 shows how colors are mapped into the YCbCr color space. The color space looks like a cube with (starting from the top in clockwise order) blue, magenta, red, yellow, green, cyan and white (in the middle) in the visible corners. Black is located on the backside, in the opposite corner of white.

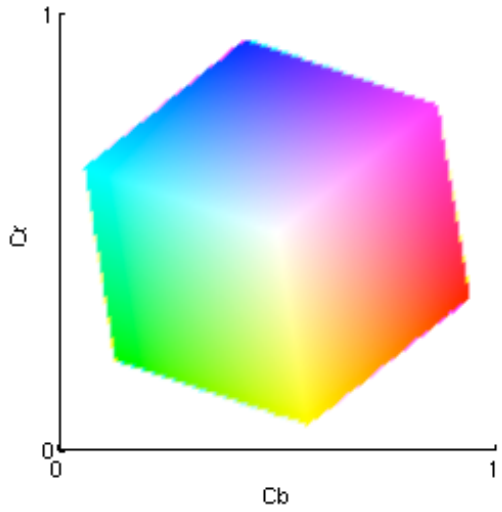


Figure 3: The YCbCr color space. The Y-axis points out of the figure.

[9] See for example http://en.wikipedia.org/wiki/YCbCr (2005-06-06).

[10] rgb2ycbcr

3.2 Low-level features

This section contains a description of the low-level features used in our model. These features are computable, in the sense that they are generated from an image by an image-processing algorithm. It is not claimed that these computable features are the only important visual properties or that they are the major factors in determining where a person will look when watching a video sequence. On the contrary, it is strongly believed that there are high-level features (that may not be visual, such as context) that are extremely important for directing gaze. The motivation for using low-level features is that they are easily computable (high-level features may be very difficult to model) combined with the results that they do correlate with gaze patterns in a significant way (Reinagel & Zador, 1999; Privitera & Stark, 2000; Itti, 2005).

There are two possible reasons that low-level features seem to correlate with gaze points. The first is that it could actually be the low-level feature in itself that influences where a person directs the gaze. As an example, this could, hypothetically, be true if motion throughout evolution has been a consistently good indicator of where to look to receive important information (such as if someone is attacking you). Then it could have become encoded in our genes that it is adaptive to look where there is motion.

The other possible reason is that the low-level feature in itself is of no importance, but it happens to correlate with an important higher-level feature. It can be discussed if a face is a high-level feature or not, but from a computational standpoint it is quite complicated to construct a general face detector (for an example, see Schneiderman, 2000). On the other hand, it is easy to detect skin color. Skin color is not a perfect face detector, but at least skin color should correlate with the existence of a face. Since it is known that people tend to look at faces (Gullberg & Holmqvist, 1999), a low-level feature that detects skin color could in fact prove useful, even though humans probably do not have a natural tendency to look at skin colored objects (unless it actually is skin).

Which of these reasons that explains the correlation is not investigated further in this thesis. But the above discussion is useful to keep in mind for the presentation of the low-level features.

3.2.1 The intensity feature

The intensity feature is simply calculated as the intensity in each of the pixels of the image. There are two variants of this feature: The first is the Y intensity value, the luminance in the pixel. The other is the Cb and the Cr values that together form a two-dimensional feature vector. In mathematical notation, this can be expressed as:

$$I_Y\left(p_{i,j}\right) = (1,0,0)\cdot p_{i,j}$$
$$I_C(p_{i,j}) = \begin{pmatrix}(0,1,0)\cdot p_{i,j}\\(0,0,1)\cdot p_{i,j}\end{pmatrix}$$

where $p_{i,j}$ is the pixel at coordinates (i,j). It is a point in the YCbCr color space and $I_Y(p)$ and $I_C(p)$ are the luminance intensity value and the color intensity values of the point.

3.2.2 The edge feature

The edge feature is calculated as the difference in intensity between adjacent pixels. Specifically, it takes the pixel intensity multiplied by 4 and subtracts the intensity of the pixels that are above, below, to the left and to the right. It then takes the absolute value. Just as for the intensity feature, there are two different variants of the edge feature. One is calculated on the Y channel, and the other is calculated on the color channels to form a two-dimensional feature vector. In mathematical notation:

$$E_Y\left(p_{i,j}\right) = \left|(1,0,0)\cdot\left(4p_{i,j} - p_{i-1,j} - p_{i+1,j} - p_{i,j-1} - p_{i,j+1}\right)\right|$$
$$E_C\left(p_{i,j}\right) = \begin{pmatrix}\left|(0,1,0)\cdot\left(4p_{i,j} - p_{i-1,j} - p_{i+1,j} - p_{i,j-1} - p_{i,j+1}\right)\right|\\\left|(0,0,1)\cdot\left(4p_{i,j} - p_{i-1,j} - p_{i+1,j} - p_{i,j-1} - p_{i,j+1}\right)\right|\end{pmatrix}$$

An example of the E_Y feature calculated for a whole image, was shown in figure 1, section 1.3. In the actual implementation, the edge feature is calculated by a convolution between the image and the matrix

$$M_{Edge} = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

followed by taking the absolute value of the resulting image. Since the resulting image is two pixels larger in height and width, the image is also cropped to the original size. The E_Y feature is then extracted as the first layer in the image, and the E_C feature is extracted from the second and the third layers.

This edge feature is a high pass frequency filter. It suppresses image content with low spatial frequencies (intensity that changes slowly) and emphasizes image content with high spatial frequencies (intensity that changes rapidly). It is suspected that high frequency correlates with high information density and, in the end, gaze points.

3.2.3 The motion feature

The motion feature is a bit more complicated than the two previous features. There are numerous different techniques and image processing algorithms that can estimate the motion in a point, given a few consecutive frames. The good ones however, do tend to get quite mathematical (see for example Brox et al). Since the motion estimation problem lies outside the scope of this thesis, our model will use an approximation. The motion feature will be approximated by *disturbance fields*, which are weighted frame differences. Disturbance fields are commonly used in motion detection algorithms, which suggests that this approximation may be apt. The disturbance field is implemented using a recursive filter, as defined by (Halevy & Weinshall, 1998), with the small modification that the absolute value is used instead.

$$\begin{aligned} A_0 &= 0 \\ A_t &= (1-w)P_t + wA_{t-1} \\ D_t &= |P_t - A_{t-1}| \end{aligned}$$

P denotes the image matrix, A_t denotes the temporal average image at frame t, and D_t is the disturbance field at frame t. The weighting factor w controls the duration for which a frame influences future values of the disturbance field. More recent frames are always given more importance, so setting w is a matter of how quickly the disturbance field "forgets" a frame difference. The value w = 0.6 was used, which according to (Trucco et al, 2002) implies that only the 6 most recent frames are expected to make a significant difference[11]. When the disturbance field is initially calculated, the 20 preceding frames are used as an initialization sequence.

Just as for the intensity and edge features, the motion feature has a Y channel variant, and a color channel variant. They are simply calculated by taking the values of the disturbance field in the corresponding point. If $d_{i,j}$ is used to denote the disturbance field at coordinates (i,j), then this can be expressed as:

$$D_Y\left(p_{i,j}\right) = (1,0,0)\cdot d_{i,j}$$
$$D_C\left(p_{i,j}\right) = \begin{pmatrix}(0,1,0)\cdot d_{i,j}\\(0,0,1)\cdot d_{i,j}\end{pmatrix}$$

Note the difference between D_Y and D_C, which are features, and D_t, which is the disturbance field at frame t.

3.2.4 The contrast feature

The contrast feature is here defined as the standard deviation in a patch, with a given size and shape, of an image. The patches that are to be used will be introduced in the next section. Unlike the other features, the contrast feature is used in three variants, one for each channel, which are all one-dimensional. The main reason for using one-dimensional color variants is that there will always be a lot fewer measurements of the contrast feature

[11] The number is probably higher here since our definition of a significant difference is smaller, as follows from the histogram intervals defined in the next section.

since it can only be calculated for patches and not for individual pixels. The mathematical definition used here, is:

$$s_\Gamma = \sqrt{\frac{\sum_{p_{i,j}\in\Gamma} p_{i,j} - \bar{p}_\Gamma}{n-1}}, \qquad \begin{aligned} C_Y(\Gamma) &= (1,0,0)\cdot s_\Gamma \\ C_{Cb}(\Gamma) &= (0,1,0)\cdot s_\Gamma \\ C_{Cr}(\Gamma) &= (0,0,1)\cdot s_\Gamma \end{aligned}$$

Here Γ denotes the image patch where the feature is calculated and $\bar{p}_\Gamma$ is the mean intensity in the three channels in the patch. The square root is taken separately in the three dimensions.

Like the edge feature, contrast is suspected to correlate with information density. Note however that there is a difference between the two features: The contrast feature disregards the relative location of the intensity variations in the patch, whereas the edge feature does not.

4 Analysis

In the analysis phase, our model is trained using eye-tracking data and feature calculations on the corresponding video sequence. The goal of the analysis is to find a description of how these features correlate with gaze points.

Until now, the word correlation has been used in a way that demands some clarifications: First, it has been discussed whether there is a correlation between gaze points and features. Strictly, this does not make sense, since correlation is defined mathematically as a value that describes the amount of co-variation between two datasets.
What is really meant by the expression, is the question whether the features are significantly different in *regions* near gaze points (regions of interest, or ROI:s) compared to other regions which are not near gaze points (or in the image as a whole).

Second, correlation has been used to characterize a general relation of dependence between gaze points and features. In other words, every kind of dependence, or co-variation, between gaze points and feature values has been described as a correlation,

with no differentiation between different kinds of co-variation. However, the strict mathematical concept of correlation can only handle linear co-variation. This means that there can be a (non-linear) co-variation between gaze points and features, with zero correlation. As an example of when this could become a problem, imagine that people often look at regions in images where the edge feature is very high, and that they also often look at regions where the edge feature is very low. If the standard correlation concept is used, this pattern will not be detected because it will seem that people in average look at regions where the edge feature takes a value somewhere in the middle.

The usage of standard correlation becomes very problematic when it comes to investigating the co-variation of gaze points and color, since it is not natural to use order relations on color (should magenta be attributed a "higher value" than blue just because it is represented by a higher value in the Cb channel?). If some colors, like skin color, are popular to look at, then that is the kind of information that should be provided by a co-variation concept, not that people on average look at colors that are a somewhat lower/higher in the color scales than the average image colors.

Previous studies on feature properties of regions have used the standard correlation concept (e.g. Reinagel & Zador, 1999). This could be enough if the purpose is just to discover that there are systematic differences in features in regions near gaze points compared to other regions. However, if prediction is the purpose of the analysis, it would be unwise to not consider non-linear co-variations. Instead of using the standard correlation, our model therefore counts the occurrence of different feature values in order to estimate their relative frequency – near gaze points as well as for the whole image. These frequency estimations are then used to construct *feature density maps*, which are estimates of the likelihood that a point is near a gaze point, given the feature value.

The proposed feature density maps contain, for each feature value, a number between zero and one. A high value indicates that the feature value has occurred disproportionally many times near a gaze point. A low value indicates that the feature value seldom occurs near gaze points. The sum of all values in the feature density map is always set to one.

4.1 Method

The video sequence was processed, one frame at a time. The features were first calculated on the whole frame. Gaze points were read from a data file (two points per subject and frame) and ROI masks were created by adding of patches of ones into a zero matrix of image size. The patches were circular and centered on the gaze points. Four ROI masks were created, each with a different patch size. Patches with radii of 15, 30, 45 and 60 pixels, corresponding to 0.5°, 1°, 1.5° and 2° of visual angle respectively, were used. Since the patches were simply added to the ROI masks, there were overlaps. Theoretically the ROI masks could therefore take integer values in the interval [0, 2n], where n is the number of subjects.

The calculated feature values were used to create histograms[12] of the occurrences of the feature values (i.e. the feature values were counted based on which bin they belonged to). This was first done in the whole frame, then in the patches. Two-dimensional features were put in three-dimensional histograms. A feature value was counted as many times as the corresponding ROI mask value. For every feature, this resulted in one histogram for the whole frame and four histograms for the ROI:s (one for each patch size). Table 3 contains information about the structure of the histograms: what intervals of the feature values that were covered, and the size of the bins. For two-dimensional features, the intervals and bin sizes are the same along both dimensions. See figure 4 for examples of histograms.

[12] According to http://en.wikipedia.org/wiki/Histogram (2005-06-06), "a histogram is the graphical version of a table which shows what proportion of cases fall into each of several or many specified categories." The term histogram will also be used to describe the actual tables as well. A histogram bin is a specific sub-interval in a histogram that constitutes a category. In a histogram, each bin is represented as a bar whose height is proportional to the relative frequency of cases falling into the category.

Histogram feature intervals		
Feature	**Interval**	**Bin size**
Intensity	[0, 1]	0.01
Edge	[0, 4]	0.01
Motion	[0, 1]	0.01
Contrast	[0, 0.1]	0.01

Table 3: Histogram structures

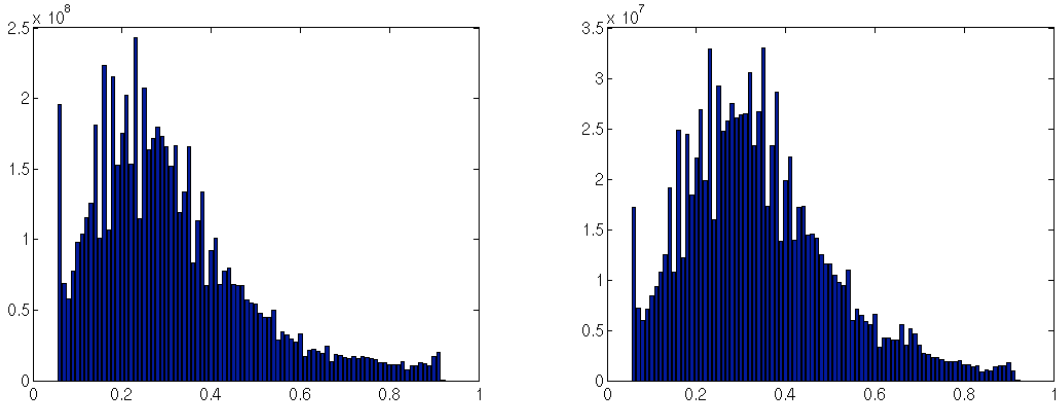


Figure 4: Histograms for ROI:s and whole frames for the I_Y feature in video sequence B.

The contrast feature was not calculated for the whole frame, but instead for patches randomly located in the image. The number of random patches was the same as the number of gaze points. This was done to facilitate a subsequent step where several non-ROI feature values were necessary. Taking the contrast on the whole image would only have given one value per frame[13]. By this procedure, the contrast measure was also calculated on regions with sizes of similar order (they are not always equal in size because some patches are located near the borders of the image).

When all the frames had been processed, the histograms were added across frames. In order to get the relative frequency of the feature values, the histograms generated from

[13] Contrast is a bit problematic due to that it is not calculated in individual points. For the other features, a value in a ROI is certain to exist also in the whole screen. This is not true for contrast.

gaze points were divided (point wise) by the histograms generated from the whole frame. The contrast histograms were divided by the random histograms generated by the same patch size. To avoid dividing by zero, one was added to the denominator in all divisions. The feature density maps were finally obtained after a normalization (the sum of each map was set to one).

4.2 Results

The feature density maps and histograms discussed in this section have been generated from both video sequence A and video sequence B. Due to practical reasons, they are presented in graphical form only. All 36 feature density maps can be found in the appendix, section 7.1. Do note that these feature maps have **not** been used for prediction, since that would violate the principle of separating the training and validation data.

The analysis hypotheses states: *There are significant differences in low-level features between regions that are looked at and regions that are not looked at.*

If there were no correlation between gaze points and the low-level features, the feature density maps would all be approximately equal to uniform probability functions (i.e. they would be almost flat). It can be seen, however, that the maps are not very regular. These irregularities indicate that some feature values more often are located near gaze points than are other feature values. For example, look at the feature density maps for I_C, the color feature. They seem to contain high values near the edges of the color space, representing extreme values of Cb and Cr. These values correspond to the pure colors in the corners of the YCbCr color space (see figure 3). Thus, it might be concluded that the subjects on average preferred to look at regions that contain pure colors, rather than mixed.

The statement of the hypothesis can be operationalized into statistical tests as follows. The objective is to prove that the feature density maps are different from uniform probability functions. This corresponds to the feature values calculated in the ROI:s and the features values calculated in the whole frame being drawn from different probability

distributions. If the two distributions, the ROI feature value distribution and the "frame total" feature value distribution, would be identical, then the feature density maps should be nearly uniform. The feature value histograms, which were used to calculate the feature density maps, can be seen as estimations of the relevant distributions. So in order to test if the distributions are different, these histograms will be used. If it is found that there is a significant difference between the histograms generated from feature values near gaze points, and the histograms generated from feature values in the whole frame, the hypothesis is taken to be true. For the contrast feature, the histograms from random points will be used instead of the histograms for the whole frame.

For the one-dimensional features, the difference will be tested using the Kolmogorov-Smirnov[14] test as well as a homogenity test[15]. For the two-dimensional features, the difference will be tested using a two-dimensional version of the same homogenity test. The results from the statistical tests are located in the appendix, section 7.3.

The Kolmogorov-Smirnov test assumes continuous data, a condition which in fact is violated since the feature values have been put in histograms with a finite number of bins. Because the feature values in themselves were not stored, they were approximated by the lower end of the bins in which they were placed (i.e. the continuous data was quantized). These violations are however not considered to have a major effect on the test (and even if they do, they ought to make it *harder* to reach significance since local variations in the data are lost).

Between 1-100 million values were calculated in the analysis phase, for each feature and patch size. A simulation-based statistical technique was applied for the Kolmogorov-Smirnov test: 10,000 quantized feature values were drawn from the empirical distributions given by the histograms, in 100 trials. The Kolmogorov-Smirnov test was applied to the drawn samples. After all the trials, the median of the test statistics and the

[14] The Matlab function kstest2 was used.

[15] A description of the homogenity test is located in the appendix, section 7.2

p-values was taken. The results can be seen in table 5. This test is very good at discovering even very small differences in distributions if the samples are large. To control that the significance was not artefactual, a control test was made where the two samples were drawn from the same empirical distribution. This control test generated non-significant p-values for all features that were tested.

The homogenity test was applied to all the feature histograms. Because of few values in certain bins, some merging of bins was necessary to fulfill the requirements for normal approximation. The number of bins used in each test can be seen in tables 6 and 7, along with the test results.

The significance levels are extremely high[16], due to the large amount of samples. The gaze point histograms are indeed different from the whole frame histograms. It also appears that the histograms for the smaller patch sizes are more deviant[17].

As an extra precaution, to see if the tests behaved correctly, some tests were also performed to see if the histograms for feature values in random regions were different from histograms for feature values in the whole frame. In those calculations, p-values were consistently nonsignificant for the Kolmogorov-Smirnov test. In the homogenity test, p-values were still extremely low, although the test statistic was considerably lower than in the real tests (about a factor of 10^4). This suggests that the test is too sensitive to small differences. However, some unpaired t-test for the sample means was also performed on 10,000 random samples. They all gave insignificant p-values, in agreement with the Kolmogorov-Smirnov test. One can also argue that the significant result in the

[16] The near-zero p-values are given by the tests because of the way they work: Given many samples, the tests will be very certain about differences between two distributions. This is reflected in the p-values. However, the difference does not need to be large to be significant. A careful interpretation of the numbers is absolutely necessary, since such small differences may well be due to a bias in the sampling data (i.e. the video sequences).

[17] Not true for the homogenity test, because more samples were available for larger patches. This was not the case in the Kolmogorov-Smirnov test, where the simulation technique was used.

homogenity test is probably due to the vast amount of samples. What these results really suggest, is that it maybe remains to find a more appropriate statistical test for this kind of hypothesis and data? The problem really lies in an assumption behind the tests, namely that the samples are representative for the whole population. If they are not, even a small bias will generate significant differences with a large number of samples.

With that in mind, it can nevertheless be concluded that the features have different distributions near gaze points compared to the rest of the frames. The hypothesis seems to hold, at least for *this* data material.

There is finally an important reservation which must be considered. There is no guarantee that the tested histograms are representative. As an example, consider that a main person in a video sequence, who is often in the gaze, typically wear clothes of a constant color (at least throughout a particular scene). This kind of regularity in objects of high interest, which may not exist in a bigger data set, gives a bias to the samples and the histograms. The deviation from uniform distributions could possibly be non-generalizable. To investigate this, the same statistical test was performed on a subset of the data, namely only video sequence A. If the suspicion of a bias is correct, then the p-values should be even smaller in these tests (if the same number of samples are used). As can be seen in tables 8, 9 and 10, the p-values do not give a clear-cut answer to this question, even if they might seem to be a bit lower for video sequence A. Also note that there are fewer samples for these tests. The suspicion of a data bias is important and needs more investigation, although this matter will not be pursued further in this thesis.

The significance levels are not undisputable, but they do indicate that these features may be used for prediction of gaze points. But the significance levels found here are not the whole truth. The ability of the features to actually predict gaze points in a highly varying set of scenes will be investigated in the next section.

5 Prediction

We are now ready to turn to the main problem of this thesis. The feature density maps, described in the previous section, will be used to predict gaze points. The prediction will be given in the shape of *prediction maps*, which are normalized to sum one and whose values (ideally) correspond to the probability of a gaze in the corresponding points.

The feature density maps generated from one video sequence are used to predict gaze points in another video sequence. The predictions are then finally validated against the eye-tracking data. The accuracy of the predictions is tested for statistical significance, first compared to random, then compared to how well a group of humans predict another group of humans.

5.1 Method

The video sequence, which is to be predicted, is processed frame by frame. The features, except for contrast, are calculated for the whole image. A grid of points is allocated to the image. For every patch size (same patch sizes as in the analysis phase), a mask is created with patches centered on the grid points. The feature values inside those patches are then gathered. Every feature and grid point is now assigned a *feature score*. The score is calculated as the sum of the values in the feature density maps, corresponding to the feature values in the patches centered on the grid point. All scores are divided by actual patch size, so that patches near the image border (which can be as small as half the original patch size) are not discriminated. To get the total score for the grid point, all the feature scores are multiplied. The frame's prediction map is finally created by interpolation[18] of the grid point scores, followed by normalization so that the sum is one. The individual feature scores of each grid point are saved, to make it possible to make new prediction maps based on any subset of the features.

[18] Any interpolation method is possible, as long as the result is non-negative everywhere. Here, spline interpolation is used (splines are piecewise polynomial functions). Negative values are then set to zero.

Since the prediction map is actually only computed based on the grid points, a large number of grid points is necessary for good prediction. Unfortunately, this also means increased computational costs (algorithm time complexity is approximately linear). Here, 99 grid points were used for the first part of video sequence B. In the second part of video sequence B, 112 grid points were used. The grid points were distributed to reflect the image aspect ratio. In the first part of video sequence B, the distance between grid points were 36 pixels vertically and 35 pixels horizontally. In the second part, distances were 55 and 51 pixels respectively.

5.2 Validation

Predictions were made on 5842 frames of video sequence B, part 1, and 6996 frames of video sequence B, part 2.[19] Some examples of predicted frames can be found in the appendix, section 7.4. Visual inspection of the prediction is of course interesting, but to make an objective estimation of the predictive ability, the accuracy has to be quantified.

The weak prediction hypothesis states: *Using only low-level features, it is possible to predict which regions will be looked at. The prediction is significantly better than what could be achieved by random.*

The hypothesis is tested as follows. First, the value of the prediction maps is evaluated in all the recorded gaze points. For each frame, the mean of these evaluations is denoted as the *model prediction score* for that frame. If the prediction would have been performed randomly, the mean of the prediction scores should be approximately one divided by the number of pixels in each frame (corresponding to a uniform distribution[20]). The hypothesis is therefore taken to be true if the mean score is significantly larger than this value – the *random prediction score*.

---

[19] Not all frames of all video sequences were predicted due to the computations being very time-consuming.

[20] It is not done here, but it would be interesting to also test the prediction against a normal distribution centered in the middle of the screen. For obvious reasons, most interesting things aren't located near the screen borders.

When the validation was made, data points were not counted where the gaze points was not registered to be in the image area. If all the gaze points were outside the image area, the frame was not included in the validation. T-tests were used to test if the model's prediction score minus the random score is greater than zero. The results are found in tables 4 and 5, under "Comparison with random".

As discussed previously, the predictor used in offline foveation is a group of humans. To really put our model's prediction capability to the test, its accuracy is also compared to that of a group of humans predicting another group. It's quite unrealistic to expect that our model will perform as good as humans, since humans logically should be the very best possible predictors of other humans. The test is rather included as a means of showing how big the difference actually is between our model's performance, and what could be demanded at most.

The strong prediction hypothesis states: *Compared to a group of humans predicting another group of humans, prediction based on low-level features is not significantly worse.*

How well can humans really predict each other's gaze points? The following approach was used to quantify the performance of a group of humans as a gaze predictor. For every frame, the subjects with gaze points outside the image were first removed. Then each individual member of the remaining subjects, one at a time, was predicted by the remaining. This was done with a (human) prediction map created by addition of Gaussians centered on the gaze points, and a normalization of the sum to one. For each subject, that prediction map is then evaluated in the gaze points of that subject. The mean of all these evaluations is taken as the *human prediction score* for the frame.

How to select the standard deviation for the Gaussians in these computations is a complicated issue. A small standard deviation gives high scores when the gaze points are close to each other, but very low scores when the gaze points diverge. A larger standard

deviation does not differentiate as much. Different standard deviations were tried, and the value that gave the highest score was used. The used standard deviation was $\sigma = 4$.[21]

T-tests were performed to see if there was a significant difference between the model prediction score and the human prediction score. The results are found in tables 4 and 5, under "Comparison with humans".

Prediction validation Video Sequence B, part 1	
Mean score, model	$1.50 \cdot 10^{-5}$
Comparison with random	
Random score	$9.86 \cdot 10^{-6}$
t-test statistic	11.62
p	$6.77 \cdot 10^{-31}$
Degrees of freedom	5841
Comparison with humans	
Mean score, humans	$1.38 \cdot 10^{-4}$
t-test statistic	35.96
p	$1.01 \cdot 10^{-268}$
Degrees of freedom	11682

Table 4: Validation, video sequence B, part 1

Prediction validation Video Sequence B, part 2	
Mean score, model	$9.74 \cdot 10^{-6}$
Comparison with random	
Random score	$4.00 \cdot 10^{-6}$
t-test statistic	45.59
p	Close to zero[22]
Degrees of freedom	6996
Comparison with humans	
Mean score, humans	$4.22 \cdot 10^{-5}$
t-test statistic	86.41
p	Close to zero
Degrees of freedom	13992

Table 5: Validation, video sequence B, part 2

In both validations, the model's prediction was better than random and not as good as human prediction. Due to the large number of samples, all differences are significant with extremely small p-values. As can be seen in figures 5 and 6, the accuracy of both the

[21] Even if this value for the standard deviation gives the highest score, it is probably not plausible for use in offline foveation. The reason is that it makes gaussians that are so thin that regions are attributed very low values in the prediction map, even if those regions are likely to be inside the areas covered by the human fovea if one looks directly at a gaze point.

[22] The value was too small for Matlab's double precision arithmetic.

model and the human prediction varies a lot between frames. Especially the model's prediction is not very robust. Actually, the model performs worse than random prediction in as many as 38% of the frames. These effects are due to varying content in different scenes. Humans ought to predict each other better when there are few regions with interesting content. In addition to that, the model also suffers from errors due to the prediction maps of different features giving high scores to unattended regions, especially in some scenes. The correlation coefficient between the two prediction scores in figure 5 and 6, is 0.43.

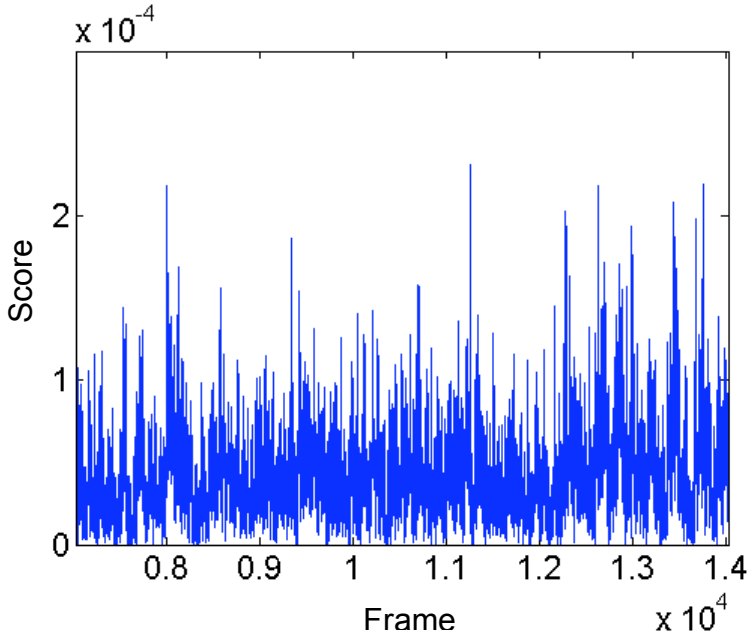


Figure 5: Human prediction score, video sequence B, part 2

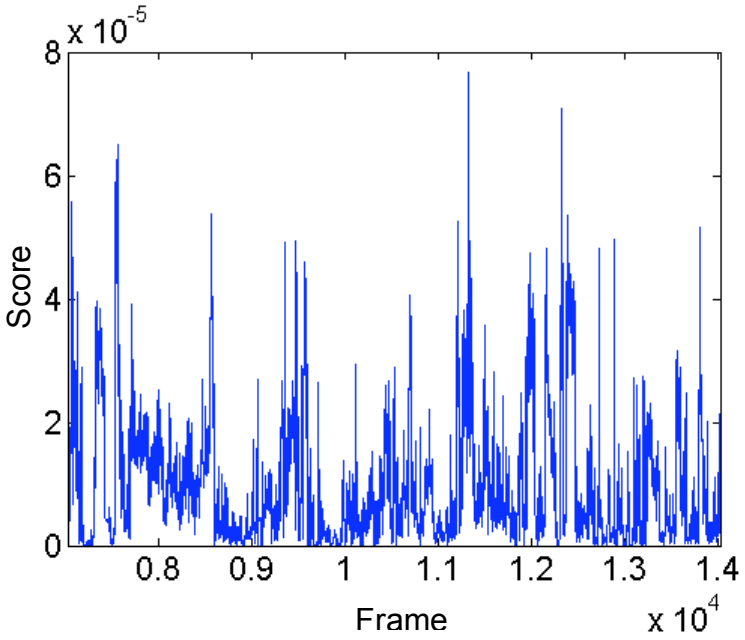


Figure 6: Model prediction score, video sequence B, part 2

To investigate the contribution of each single feature, prediction was also performed using only one feature at a time. The results from this test, which was only performed for video sequence B, part 2, can be found in table 6. By doing this type of validation, it is possible to separate the usefulness of each feature, from the part of the model where the features are combined.

Prediction validation – single feature prediction Video Sequence B, part 2	
Mean score, I_Y	$3.8 \cdot 10^{-6}$
Mean score, I_C	$4.1 \cdot 10^{-6}$
Mean score, E_Y	$4.3 \cdot 10^{-6}$
Mean score, E_C	$5.2 \cdot 10^{-6}$
Mean score, D_Y	$4.5 \cdot 10^{-6}$
Mean score, D_C	$4.4 \cdot 10^{-6}$
Mean score, C_Y	$7.2 \cdot 10^{-6}$
Mean score, C_{Cb}	$6.8 \cdot 10^{-6}$
Mean score, C_{Cr}	$8.0 \cdot 10^{-6}$

Table 6: Validation, video sequence B, part 2, prediction based on individual features

Interestingly, some of these scores are quite close to the score of the whole model, which uses all the features. This suggests that the multiplication of feature scores used by the model is not a very good approach. There is probably useful information hidden in the features, that the model does not take account of. This may possibly be because multiplication requires that none of the feature scores are low, in order to give a high value. Another method, where the feature scores were summed in order to create the prediction maps, was also tried. Both simple summation of the feature scores, as well as a weighted summation, where the mean scores in table 6 gave the weights, were tried. The results are found in table 7. As can be seen however, summation does not appear to be better than multiplication in this case.

Model prediction scores – feature summation Video Sequence B, part 2	
Mean score, model using summation	$6.87 \cdot 10^{-6}$
Mean score, model using weighted summation	$6.94 \cdot 10^{-6}$

Table 7: Model prediction scores, using feature score summation

6 Discussion

The results indicate that the selected features strongly correlate with gaze points of human video observers. There really seems to be major differences in low-level features between attended and unattended regions in the video frames. One has to be careful however, in how to interpret these differences. For example, it seems from these results that people generally prefer looking at regions where there are pure colors, and where there is high contrast. But the true cause of these tendencies is yet to be revealed. They could be due to some general preference to look at such regions, that people have because of one of the two explanations given in section 3.2. But they may in fact also be due to biases in the used data material. I do argue that all of the irregularities in the feature density maps found here hardly are artefactual, but the shapes of the maps may be specific to the used data material.

A disadvantage of building a model, such as this, based on experimental data, is that it is contextual. If the data material on which the model is built is non-representative, the feature statistics may not be generalizable. The interesting question is if there is such a thing as a representative video sequence? Since there are so many aspects that can be considered, the answer is probably no in general. But one has to remember that it is the low-level feature values that are used, and estimations of their distributions may very well be applicable to a wide range of data, even though there will always be exceptions, of course. But perhaps there is a common denominator for low-level feature distributions, for a large class of video sequences? A more thorough investigation in this matter could probably shed some light on the issue. A possible approach to testing the generalizability would be to compare feature value distributions generated from several different data materials. If they turn out to be very different, then there is a problem.

The level of predictional accuracy of our model lay, as expected, somewhere between those of random and human prediction. Still, there is no point in trying to conceal that the author had hoped that our model would perform better. It should not be hard to beat a uniform random prediction. It should actually be enough to use the fact that most people look at the center of the screen during most of the time. Therefore, a predictor which is not able to beat that score in 38% of the frames, is nowhere as strong and robust as it has to be.

Concerning the strong prediction hypothesis, it has to be remembered that it is a very strong statement. A model which is as good as humans in predicting where other humans will look, in *all types of scenes*, might never see the light of day. But the gap in predictive ability is most likely a difference which is possible to shrink. The strong prediction hypothesis has not been met here, but it has in no way been proved that it is impossible to do so in the future, at least for some specific types of scenes. Gaze prediction is still in its early stages, and is most likely not a futile endeavor after all.

The proposed approach, where the relative frequencies of feature values are used for prediction, is still interesting in the views of the author, and should not be discarded on

the grounds of the predictive accuracy attained here. The approach offers a way of leveraging on feature values, which has not been considered previously. It lets the model consider a more general class of correlation between gaze points and features, removing some of the limitations that follow from arranging feature values on a scale. This is an advantage over classical methods.

Of course, a lot more development and testing have to be made in order to benefit from this approach. A starting point could be in the combination of the feature scores. The results indicate that simple multiplication and addition are too crude. Perhaps a neural network approach would work better? The results from prediction based on single features also indicate that movement does not give as high accuracy as was expected. Thus, the selected approximation of movement (disturbance fields) is probably not a good one.

The strange rectangular pattern that the prediction model gives (see the examples in 7.4) is most likely artefactual. It may be due to certain features giving low scores in some regions, and thereby generating low scores for those grid points. The interpolation method then gives low values to whole regions. This problem could possibly be fixed by changing the interpolation method, how the feature scores are combined, or possibly by using more grid points.

A major problem in gaze prediction based on low-level features, is the things that the features are incapable of discovering. Examples of such things are expectations: If it is expected that someone will walk out of a car, how could a predictor based on low-level features possibly know that? Events of that kind are likely to require the model to include top-down aspects as well.

Nevertheless, the features used here are just a small selection of the endless possibilities available. Everything that can be generated by an image-processing algorithm can be used as a feature. And, as discussed recently, with the relative frequency approach, the feature values does not even have to be arranged in a scale. This means that it is possible

to include “features” that handle special cases, like a face or text detector. There is also room for improvement of the existing features. Motion has already been mentioned – a better motion detection algorithm would almost certainly increase the predictive ability of the model. A type of situations that probably need special care are scenes where the camera is moving. As the motion feature works now, this gives a high value for the whole frame, while human viewers might hardly even notice the movements, if they are focused on an object that is relatively still (relative to the camera).

More added features unfortunately bring a problem with them. As the number of features is increased, the risk of overlapping increases. As an example, consider a face detector being integrated into the model. They way that it is constructed here, the data analysis would still give relatively high values to skin colors – even though the additional predictive strength of skin colors is diminished due to the face detector. This phenomenon introduces a bias with increased number of features. One has to be careful not to add several features that cover the same aspect and thus correlate with each other. Maybe it is possible to integrate a mechanism in the model that removes such effects? Such a mechanism could, for example, be a “winner-takes-all” function, where the feature that has the highest predictive ability at any time, does the prediction by itself. Maybe neurobiological research could provide a starting point for a solution to the problem? A purely statistical approach, in the spirit of this thesis, would be to use multidimensional, joint histograms that cover several features in common distributions. That solution, however, increases the need for (representative) data.

Other possible improvements of our model include quantitative, such as using more patches (of other sizes), a finer grid, or a smaller bin size for the histograms. More complicated possible improvements include trying different model structures. For example it would be interesting to see how well a boosting algorithm, such as the one used by (Ericsson & Pehrsson, 2005) can perform for moving images. If a scene change detector would be included in the model, it would be possible to use different prediction structures (or feature density maps) for different types of scenes. Although first it should

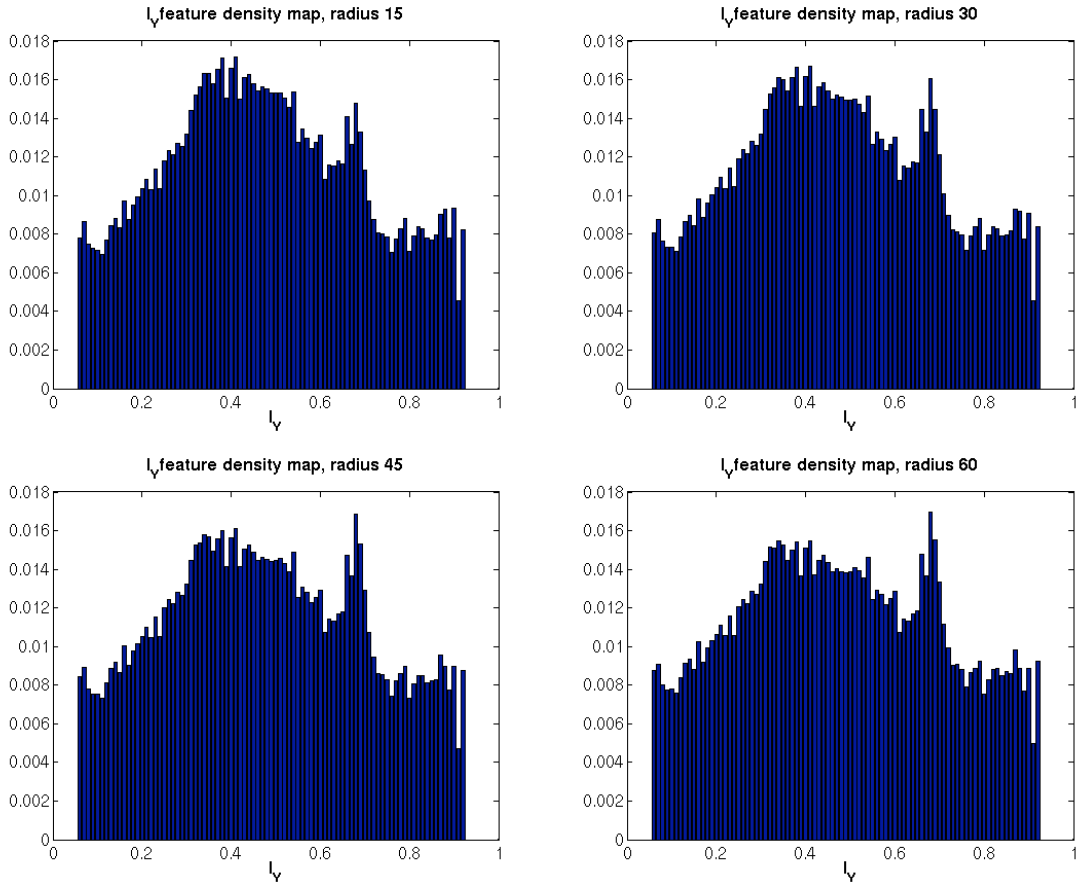
be established that it is possible to categorize scenes in a general way, that makes sense to the model (i.e. makes the scene classification useful in terms of improved prediction).

In conclusion, there seems to be many years left of research before gaze prediction can become useful for applications. Still, the findings that certain features tend to be different in attended regions, is interesting from a theoretical standpoint. Regarding this attempt of gaze prediction, it has initially not delivered any results of quality. But although the achieved accuracy indicates a failure, there is still hope that the selected approach can prove to be rewarding in the future. Thus the failure is considered an interesting one.

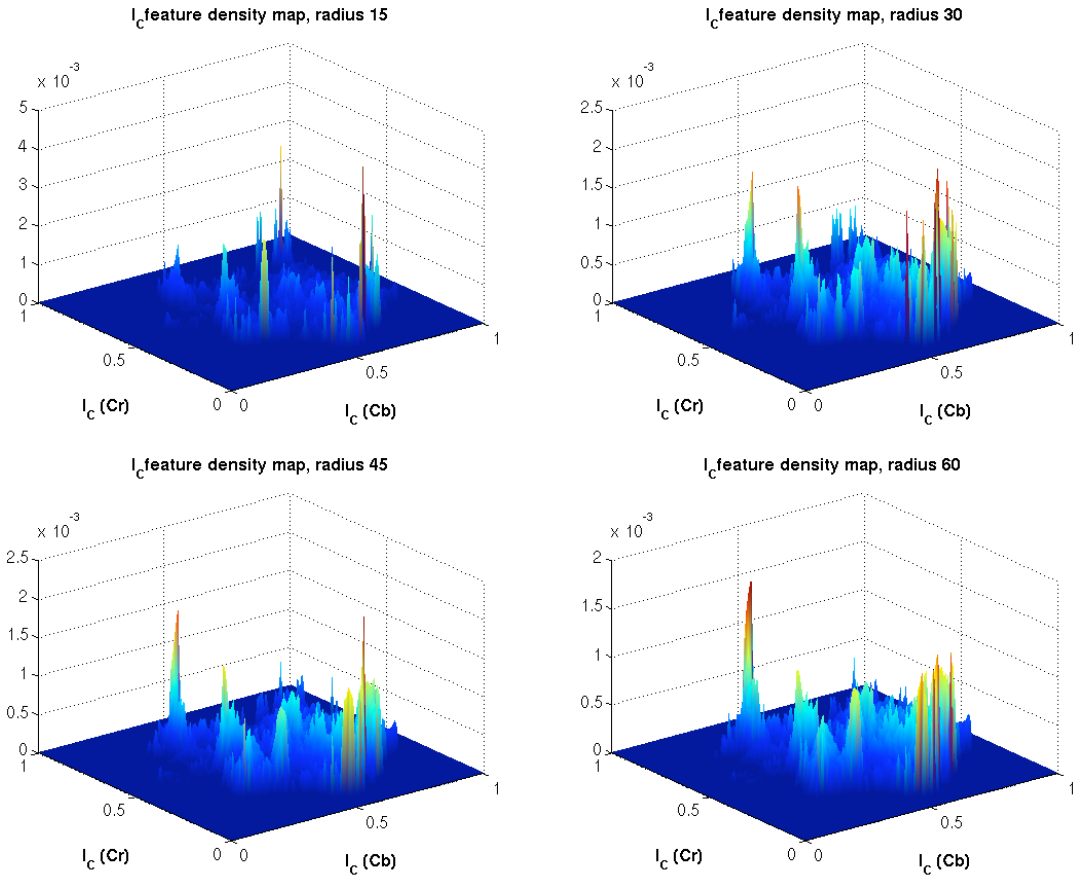
7 Appendix

7.1 Feature density maps

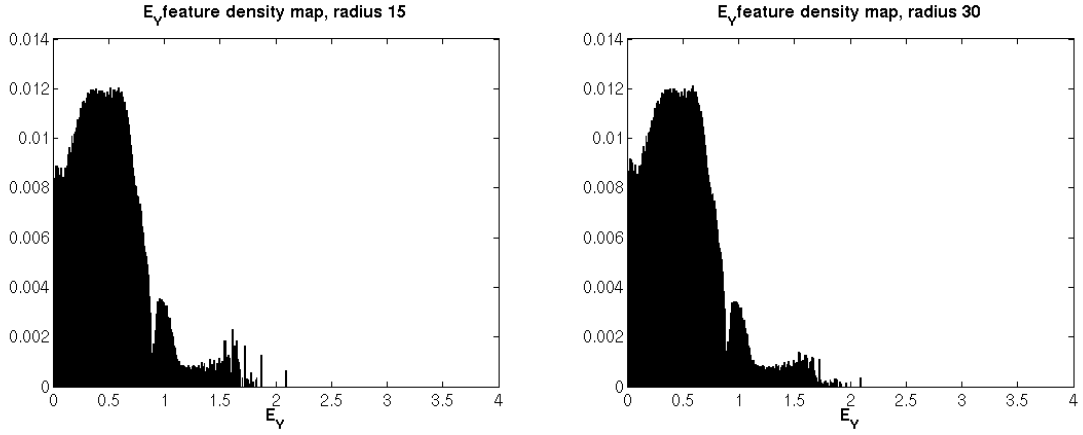
This section contains graphical representations of the feature density maps generated by the analysis described in section 4. The feature density maps shown here have been generated from both video sequence A and video sequence B.



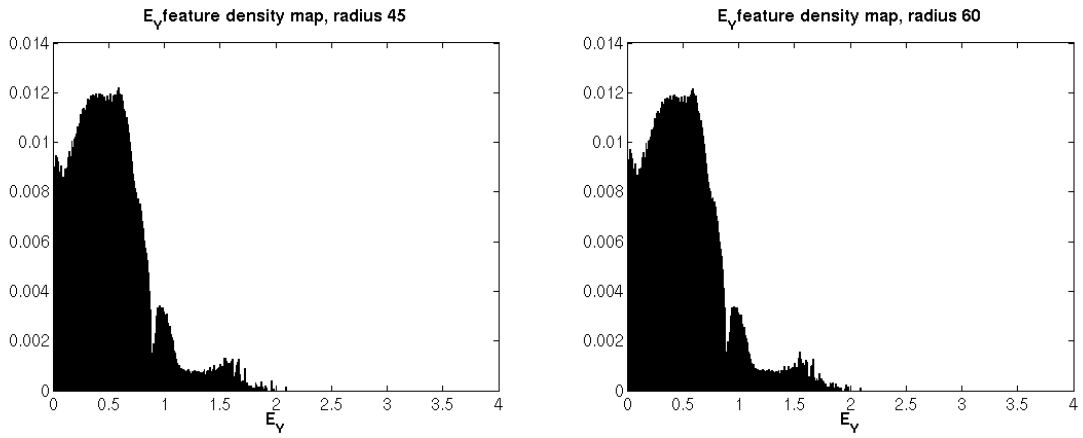
Figures 7-10: Feature density maps, I_Y feature



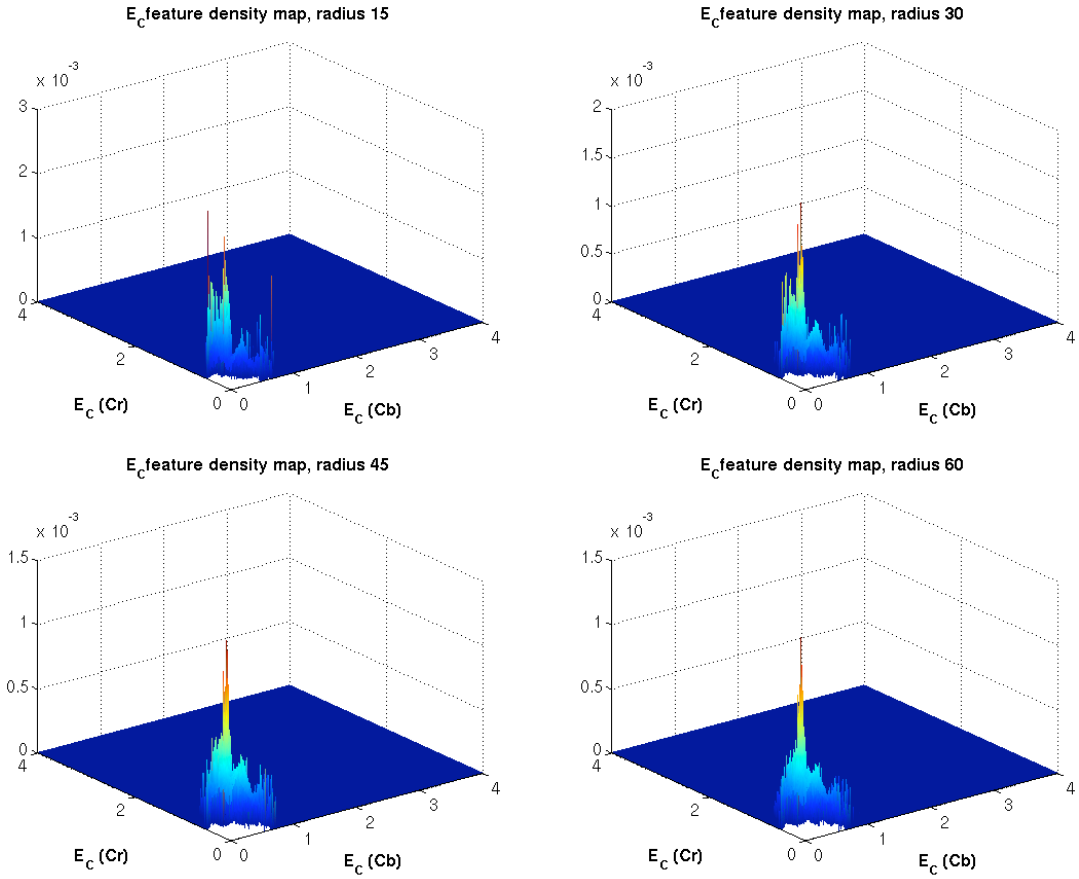
Figures 11-14: Feature density maps, I_C feature



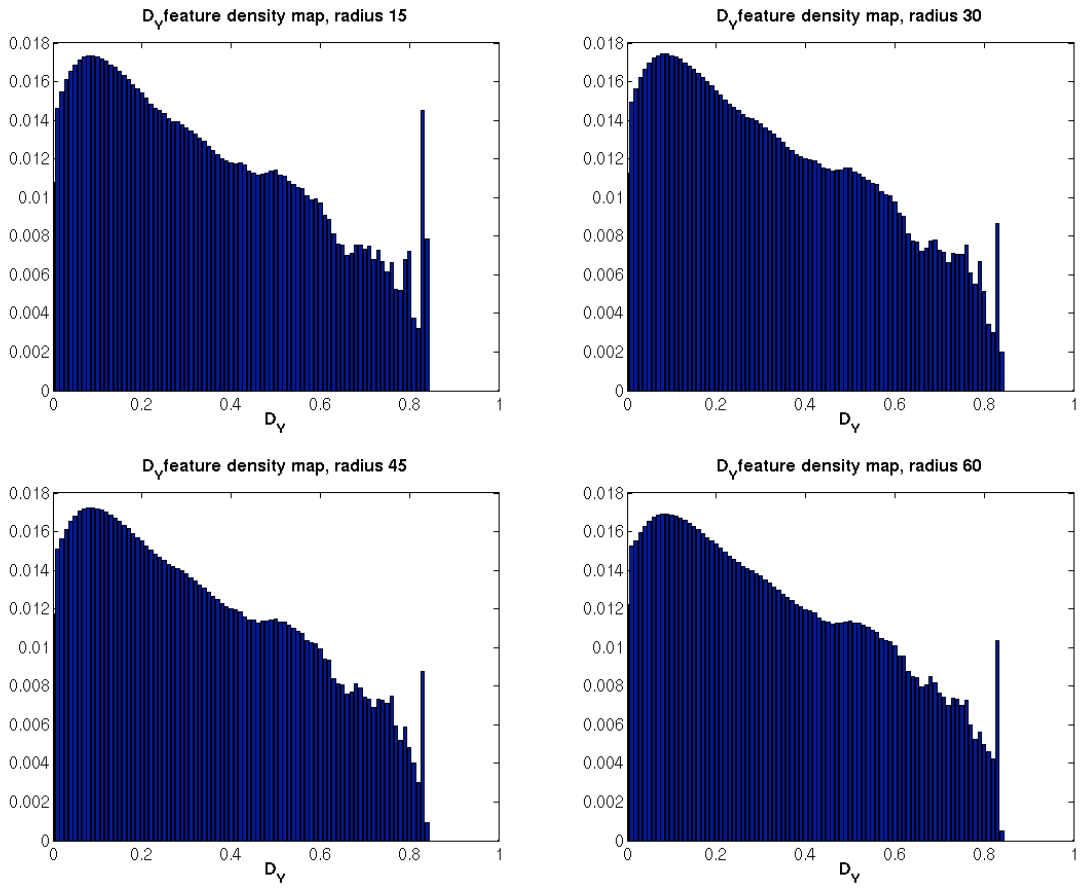
Figures 15-16: Feature density maps, E_Y feature



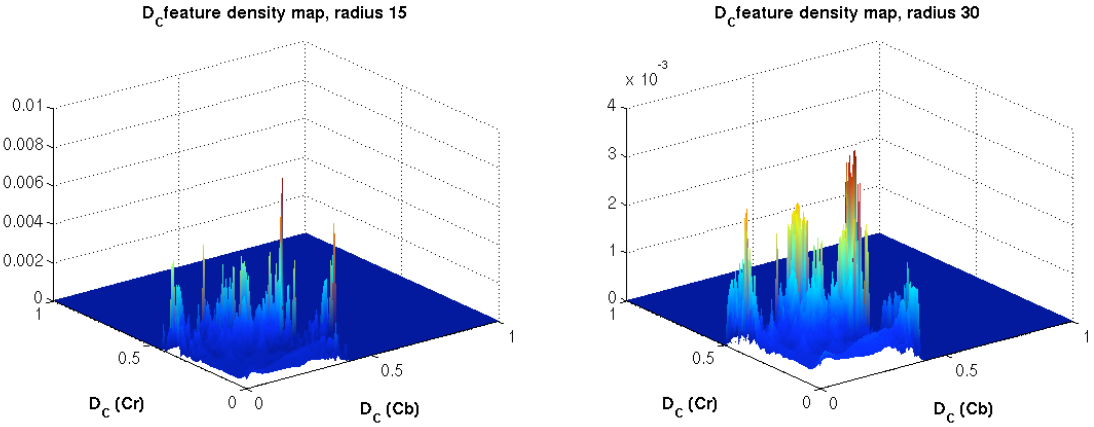
Figures 17-18: Feature density maps, E_Y feature



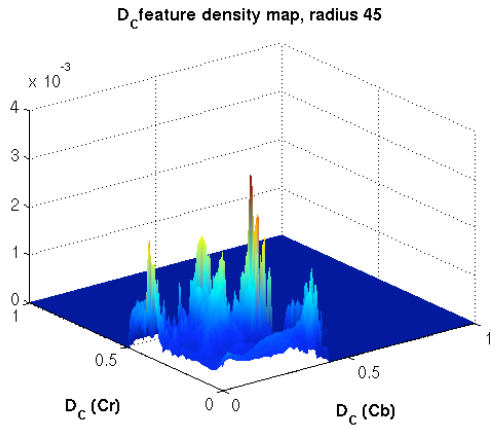
Figures 19-22: Feature density maps, E_C feature

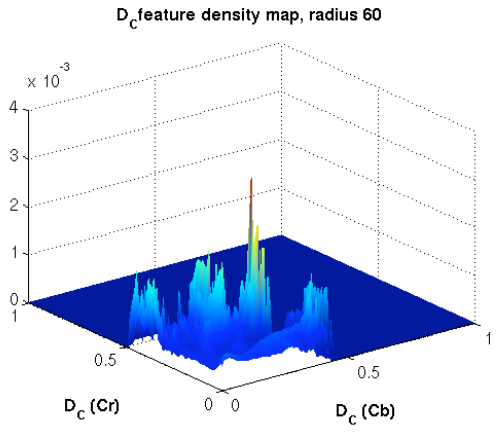


Figures 23-26: Feature density maps, D_Y feature

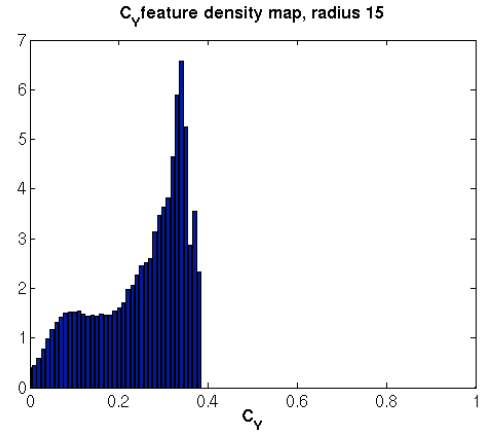


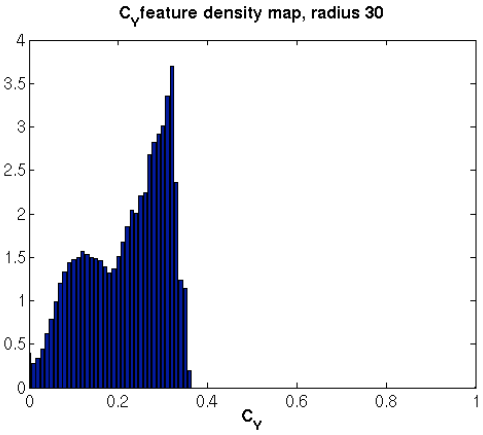
Figures 27-28: Feature density maps, D_C feature

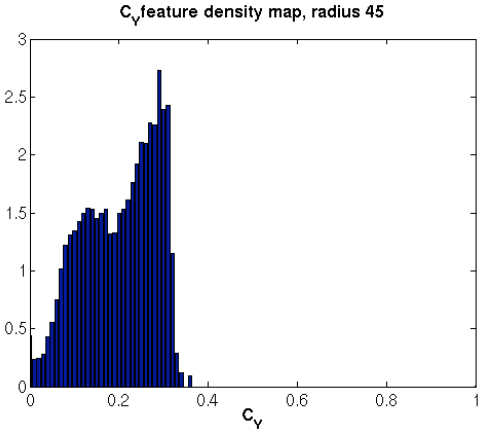


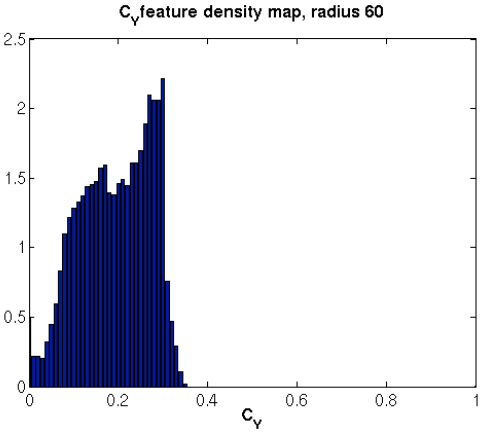


Figures 29-30: Feature density maps, D_C feature

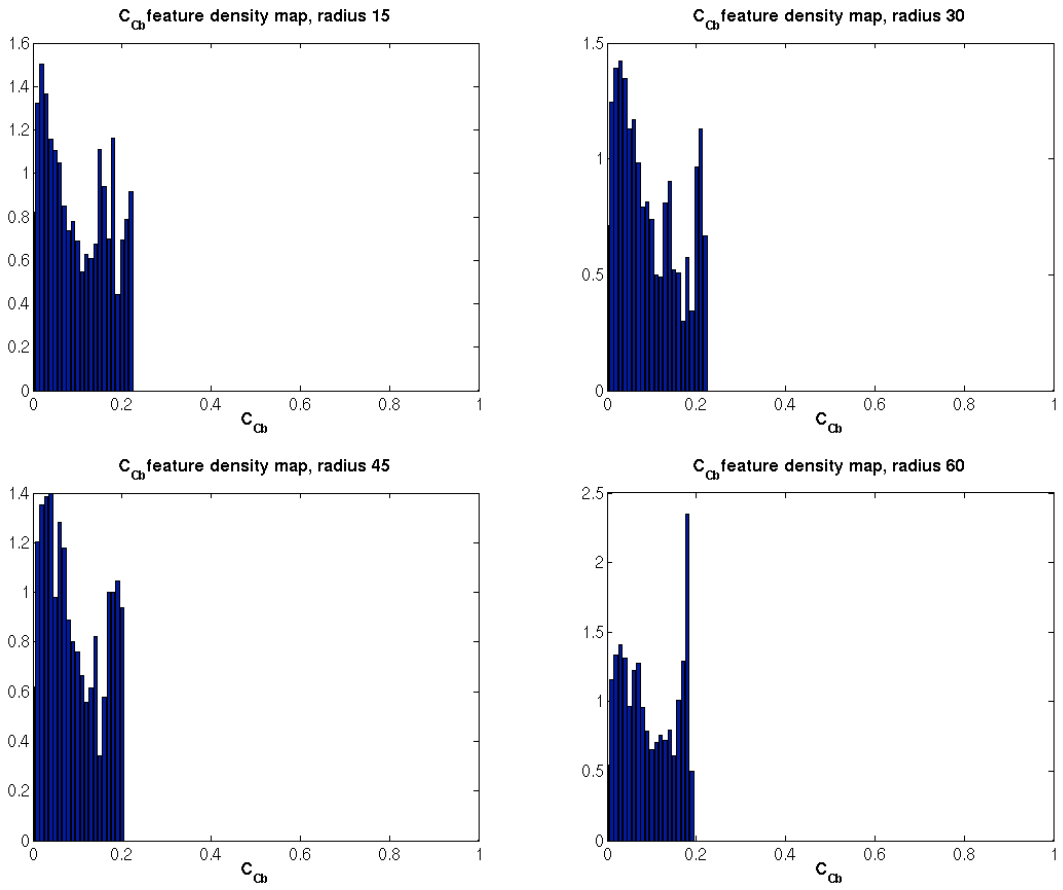




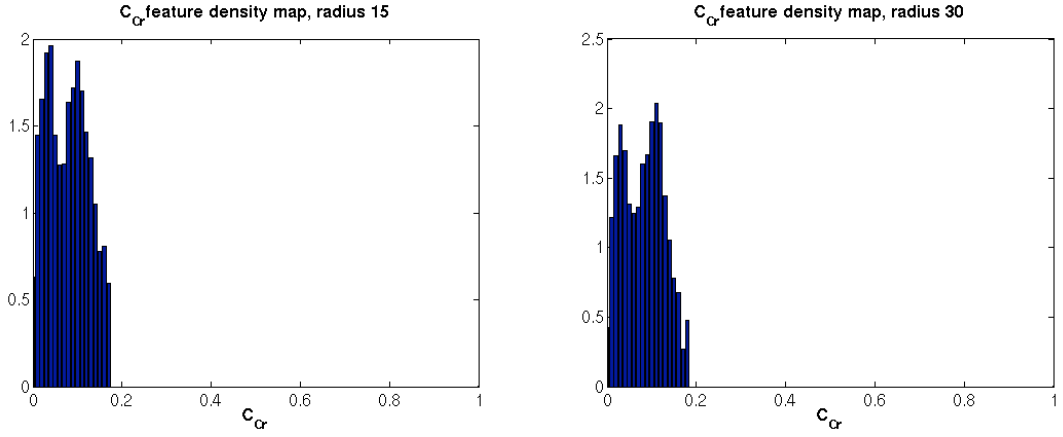




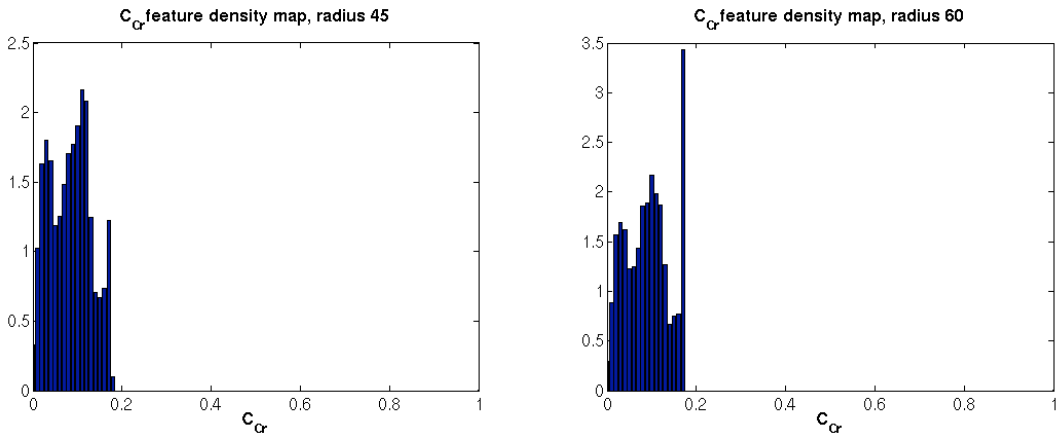
Figures 31-34: Feature density maps, C_Y feature



Figures 35-38: Feature density maps, C_{Cb} feature



Figures 39-40: Feature density maps, C_{Cr} feature



Figures 41-42: Feature density maps, C_{Cr} feature

7.2 The homogenity test

The object of the test is to see whether two histograms are generated by different distribution. The test can be used for distributions of any dimension. Let H_0 be the hypothesis that the distributions are equal. We have two samples of sizes N and M respectively. The number of samples in each bin is denoted by n_k and m_k, where k is the bin index. If H_0 is true, then an estimation of the probability of a sample in bin k is

$$p_k = \frac{(n_k + m_k)}{N + M}$$

Now,

$$T_n = \sum_k \frac{(n_k - Np_k)^2}{(Np_k(1-p_k))} \quad \text{and} \quad T_m = \sum_k \frac{(m_k - Mp_k)^2}{(Mp_k(1-p_k))}$$

are approximately χ^2-distributed and can be used to test H_0. However, this approximation might be crude. A further refinement can however be done: The distribution of the number of observations n_k, conditioned on the total number of samples in the bin, $n_k + m_k$, is of Bin($n_k + m_k$,N/(N+M)), given that H_0 is true. (We are still neglecting the dependence between boxes due to the finite number of samples.) Now, we have that

$$T = \sum_k \left(\frac{(n_k - Np_k)}{N + M} \right)^2 \Big/ \frac{NMp_k}{(N+M)^2}$$

is approximately χ^2-distributed. To get the significance level at which H_0 can be rejected, the right quantile of the χ^2-distribution is evaluated at T (χ^2-test). The number of degrees of freedom is the number of bins minus one.

As a rule of thumb, the normal approximation is viable if the expected number of samples in each bin is more or equal to 5 under H_0. This is equivalent to the bins being so large that

$$\min_k \left(p_k \cdot \frac{\min(N, M)}{N + M} \right) > 5 .$$

Bins with zero samples in the edges of the histograms (i.e. outside the area where the feature takes values) are ignored by this rule. The test would have given the same result if those bins had never been in the histogram. The zero-sample bins are of course not included in the degrees of freedom either.

7.3 Statistical tests of the analysis hypothesis

Kolmogorov-Smirnov test Video sequence A and B			
Feature	**Patch radius**	**Test statistic**	**p**
I_Y	15	0.0893	$<10^{-34}$
I_Y	30	0.0822	$<10^{-29}$
I_Y	45	0.0783	$<10^{-26}$
I_Y	60	0.0704	$<10^{-21}$
E_Y	15	0.0563	$<10^{-13}$
E_Y	30	0.0524	$<10^{-11}$
E_Y	45	0.0475	$<10^{-9}$
E_Y	60	0.0423	$<10^{-7}$
D_Y	15	0.0942	$<10^{-38}$
D_Y	30	0.0854	$<10^{-31}$
D_Y	45	0.0752	$<10^{-24}$
D_Y	60	0.0634	$<10^{-17}$
C_Y	15	0.2059	$<10^{-184}$
C_Y	30	0.2228	$<10^{-216}$
C_Y	45	0.2234	$<10^{-217}$
C_Y	60	0.2145	$<10^{-200}$
C_{Cb}	15	0.1164	$<10^{-58}$
C_{Cb}	30	0.1408	$<10^{-86}$
C_{Cb}	45	0.1458	$<10^{-92}$
C_{Cb}	60	0.1412	$<10^{-86}$
C_{Cr}	15	0.2212	$<10^{-212}$
C_{Cr}	30	0.2495	$<10^{-271}$
C_{Cr}	45	0.2276	$<10^{-225}$
C_{Cr}	60	0.2349	$<10^{-240}$

Table 8: Kolmogorov-Smirnov test for difference between histograms

One-dimensional homogenity test Video sequence A and B					
Feature	**Patch radius**	**Number of bins**	**Test statistic**	**p**	**df**
I_Y	15	101	$5.1038 \cdot 10^6$	Close to zero[23]	83
I_Y	30	101	$1.7003 \cdot 10^7$	Close to zero	83
I_Y	45	101	$2.9906 \cdot 10^7$	Close to zero	83
I_Y	60	101	$4.0527 \cdot 10^7$	Close to zero	83
E_Y	15	201	$2.0370 \cdot 10^6$	Close to zero	91
E_Y	30	201	$6.2707 \cdot 10^6$	Close to zero	98
E_Y	45	201	$9.8538 \cdot 10^6$	Close to zero	99
E_Y	60	201	$1.1696 \cdot 10^7$	Close to zero	100
D_Y	15	101	$4.8741 \cdot 10^6$	Close to zero	83
D_Y	30	101	$1.5990 \cdot 10^7$	Close to zero	83
D_Y	45	101	$2.7137 \cdot 10^7$	Close to zero	83
D_Y	60	101	$3.4290 \cdot 10^7$	Close to zero	83
C_Y	15	21	$1.3345 \cdot 10^5$	Close to zero	7
C_Y	30	21	$1.8079 \cdot 10^5$	Close to zero	7
C_Y	45	21	$1.8276 \cdot 10^5$	Close to zero	6
C_Y	60	21	$1.6821 \cdot 10^5$	Close to zero	6
C_{Cb}	15	51	$1.1256 \cdot 10^4$	Close to zero	11
C_{Cb}	30	51	$1.6404 \cdot 10^4$	Close to zero	11
C_{Cb}	45	51	$1.8894 \cdot 10^4$	Close to zero	10
C_{Cb}	60	51	$2.2039 \cdot 10^4$	Close to zero	9
C_{Cr}	15	26	$3.9134 \cdot 10^3$	Close to zero	4
C_{Cr}	30	26	$3.5941 \cdot 10^3$	Close to zero	4
C_{Cr}	45	26	$5.2001 \cdot 10^3$	Close to zero	4
C_{Cr}	60	26	$7.6160 \cdot 10^3$	Close to zero	4

Table 9: One-dimensional homogenity test for difference between histograms

[23] The value was too small for Matlab's double precision arithmetic.

Two-dimensional homogenity test Video sequence A and B					
Feature	**Patch radius**	**Number of bins**	**Test statistic**	**p**	**df**
I_C	15	10×10	$1.7623 \cdot 10^9$	Close to zero	45
I_C	30	10×10	$4.1442 \cdot 10^8$	Close to zero	46
I_C	45	10×10	$5.1990 \cdot 10^9$	Close to zero	47
I_C	60	10×10	$5.3038 \cdot 10^8$	Close to zero	47
E_C	15	41×41	$1.1016 \cdot 10^7$	Close to zero	72
E_C	30	41×41	$3.7506 \cdot 10^7$	Close to zero	86
E_C	45	41×41	$7.1231 \cdot 10^7$	Close to zero	96
E_C	60	27×27	$9.8890 \cdot 10^7$	Close to zero	55
D_C	15	11×11	$2.0465 \cdot 10^5$	Close to zero	24
D_C	30	11×11	$6.4649 \cdot 10^5$	Close to zero	24
D_C	45	11×11	$1.0973 \cdot 10^9$	Close to zero	26
D_C	60	11×11	$1.4615 \cdot 10^9$	Close to zero	29

Table 10: Two-dimensional homogenity test for difference between histograms

Kolmogorov-Smirnov test Video sequence A			
Feature	**Patch radius**	**Test statistic**	**p**
I_Y	15	0.0714	$<10^{-21}$
I_Y	30	0.0708	$<10^{-21}$
I_Y	45	0.0676	$<10^{-19}$
I_Y	60	0.0655	$<10^{-18}$
E_Y	15	0.0537	$<10^{-12}$
E_Y	30	0.0522	$<10^{-11}$
E_Y	45	0.0494	$<10^{-10}$
E_Y	60	0.0428	$<10^{-7}$
D_Y	15	0.1019	$<10^{-44}$
D_Y	30	0.0971	$<10^{-40}$
D_Y	45	0.0925	$<10^{-36}$
D_Y	60	0.0877	$<10^{-33}$
C_Y	15	0.1447	$<10^{-90}$
C_Y	30	0.1702	$<10^{-125}$
C_Y	45	0.1767	$<10^{-135}$
C_Y	60	0.1729	$<10^{-130}$
C_{Cb}	15	0.1492	$<10^{-96}$
C_{Cb}	30	0.1850	$<10^{-148}$
C_{Cb}	45	0.1823	$<10^{-144}$
C_{Cb}	60	0.1845	$<10^{-148}$
C_{Cr}	15	0.1986	$<10^{-171}$
C_{Cr}	30	0.2269	$<10^{-224}$
C_{Cr}	45	0.2398	$<10^{-250}$
C_{Cr}	60	0.2483	$<10^{-268}$

Table 11: Kolmogorov-Smirnov test for difference between histograms

One-dimensional homogenity test Video sequence A					
Feature	**Patch radius**	**Number of bins**	**Test statistic**	**p**	**df**
I_Y	15	101	$5.1038 \cdot 10^6$	Close to zero	83
I_Y	30	101	$1.7003 \cdot 10^7$	Close to zero	83
I_Y	45	101	$2.9906 \cdot 10^7$	Close to zero	83
I_Y	60	101	$4.0527 \cdot 10^7$	Close to zero	83
E_Y	15	201	$2.0370 \cdot 10^6$	Close to zero	91
E_Y	30	201	$6.2707 \cdot 10^6$	Close to zero	98
E_Y	45	201	$9.8538 \cdot 10^6$	Close to zero	99
E_Y	60	201	$1.1696 \cdot 10^7$	Close to zero	100
D_Y	15	101	$4.8741 \cdot 10^6$	Close to zero	83
D_Y	30	101	$1.5990 \cdot 10^7$	Close to zero	83
D_Y	45	101	$2.7137 \cdot 10^7$	Close to zero	83
D_Y	60	101	$3.4290 \cdot 10^7$	Close to zero	83
C_Y	15	34	$9.2323 \cdot 10^3$	Close to zero	12
C_Y	30	34	$1.1471 \cdot 10^4$	Close to zero	12
C_Y	45	34	$1.1972 \cdot 10^4$	Close to zero	12
C_Y	60	34	$1.2125 \cdot 10^4$	Close to zero	12
C_{Cb}	15	51	$6.3363 \cdot 10^3$	Close to zero	7
C_{Cb}	30	51	$1.0048 \cdot 10^4$	Close to zero	8
C_{Cb}	45	51	$1.1021 \cdot 10^4$	Close to zero	9
C_{Cb}	60	51	$1.0595 \cdot 10^4$	Close to zero	9
C_{Cr}	15	26	$7.7238 \cdot 10^3$	Close to zero	4
C_{Cr}	30	26	$1.2305 \cdot 10^4$	Close to zero	4
C_{Cr}	45	26	$1.5290 \cdot 10^4$	Close to zero	4
C_{Cr}	60	26	$1.7596 \cdot 10^4$	Close to zero	4

Table 12: One-dimensional homogenity test for difference between histograms

Two-dimensional homogenity test Video sequence A					
Feature	**Patch radius**	**Number of bins**	**Test statistic**	**p**	**df**
I_C	15	10×10	$2.2453 \cdot 10^7$	Close to zero	42
I_C	30	9×9	$7.1852 \cdot 10^7$	Close to zero	35
I_C	45	9×9	$1.1571 \cdot 10^8$	Close to zero	35
I_C	60	9×9	$1.4673 \cdot 10^8$	Close to zero	35
E_C	15	26×26	$9.3829 \cdot 10^5$	Close to zero	22
E_C	30	26×26	$3.4098 \cdot 10^6$	Close to zero	29
E_C	45	26×26	$6.9117 \cdot 10^6$	Close to zero	32
E_C	60	26×26	$1.1119 \cdot 10^7$	Close to zero	34
D_C	15	11×11	$3.0119 \cdot 10^6$	Close to zero	24
D_C	30	11×11	$9.2214 \cdot 10^6$	Close to zero	24
D_C	45	11×11	$1.4708 \cdot 10^7$	Close to zero	26
D_C	60	11×11	$1.7893 \cdot 10^7$	Close to zero	29

Table 13: Two-dimensional homogenity test for difference between histograms

7.4 Examples of predicted frames

Here, two examples of predicted frames are given. In each example, the original frame is displayed first, followed by the predictions of the model and that of humans. The predictions are visualized in the following manner: First, the prediction maps were calculated. The human prediction map was created (for this purpose) by adding Gaussians with standard deviation $\sigma = 30$ in the gaze points. Both prediction maps were then normalized to an average intensity of 0.3 and any values larger than 1 were set to 1. The two last steps were repeated until the average value was approximately 0.3. The sums of prediction maps were thus approximately the same. The image matrices of the frames were finally point wise multiplied by the respective prediction maps, to generate the visualization.

Figure 43: video sequence B, part 2, frame 9800

Figure 44: video sequence B, part 2, frame 9800, predicted by our model

Figure 45: video sequence B, part 2, frame 9800, predicted by humans

Figure 46: video sequence B, part 2, frame 10050

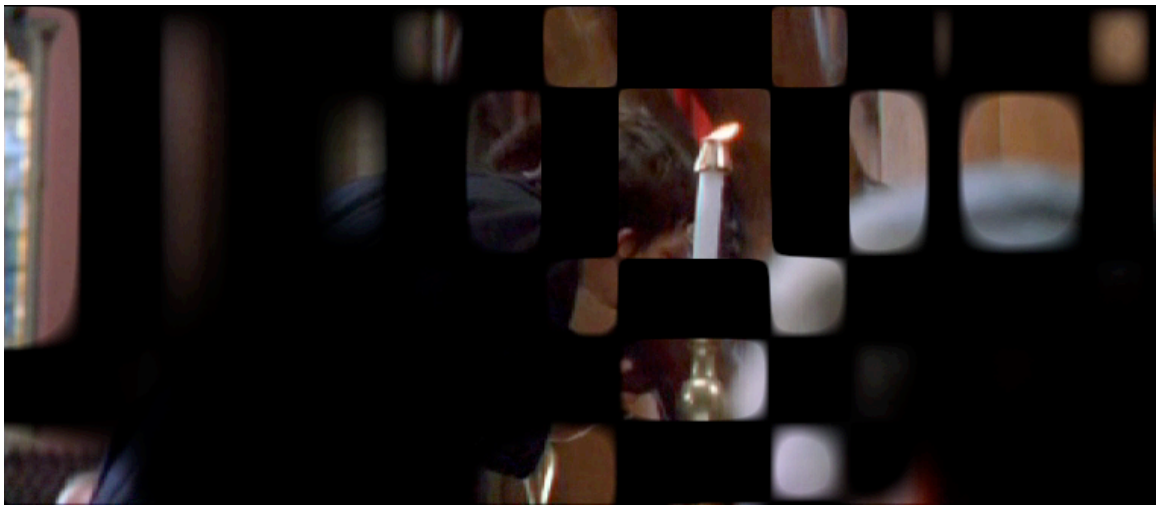


Figure 47: video sequence B, part 2, frame 10050, predicted by our model

Figure 48: video sequence B, part 2, frame 10050, predicted by humans

References

Brox, T., Bruhn, A., Papenberg N. & Weickert, J. (2004) *High Accuracy Optical Flow Estimation Based on a Theory for Warping.* In Pajdlaand, T., Matas, J. (Eds.) ECCV 2004, LNCS3024, pp. 25–36. Springer-Verlag Berlin Heidelberg 2004.

Dorr, M., Böhme, M., Drewes, J., Gegenfurtner K. R., & Barth, E. (2005) *Variability of eye movements on high-resolution natural videos.* Poster presented at TWK: 8th Tübingen Perception Conference, 25th - 27th Feb 2005.

Duchowski, A. (2000) *Introduction to the Human Visual System (HVS).* First part in course notes to Eye-Based Interaction in Graphical Systems: Theory & Practice.

Ericsson, A. & Pehrsson, M. (2005) *Boosting auto-focus object selection in mobile phone digital cameras.* Master's thesis, Lund Institute of Technology.

Goldstein, R. B., Peli, E., Lerner, S., & Luo, G. (2004). *Eye movements while watching video: comparisons across viewer groups* [Abstract]. Journal of Vision, 4(8), 643a, http://journalofvision.org/4/8/643/, doi:10.1167/4.8.643.

Gullberg, M. & Holmqvist, K., (1999) *Focus on Gesture*. Pragmatics & Cognition, 7(1), 65-73.

Halevy, G. & Weinshall, D. (1998) *Motion of disturbances: detection and tracking of multi-body non-rigid motion.* Machine Vision and Applications (1999) 11: 122–137

Henderson, J.M., Falk, R., Minut, S., Dyer, F.C. & Mahadevan, S. (2000) *Gaze Control for Face Learning and Recognition by Humans and Machines.* Michigan State University Eye Movement Laboratory Technical Report 2000, 4, 1-14.

Itti, L. (2005) *Quantifying the Contribution of Low-Level Saliency to Human Eye Movements in Dynamic Scenes*, Visual Cognition, (in press)

Itti, L. (2004) *Automatic foveation for video compression using a neurobiological model of visual attention*. IEEE Transactions on image processing, Vol. 13, No.10, October 2004.

Itti, L. & Koch, C. (2000) *A saliency-based search mechanism for overt and covert shifts of visual attention*. Vision Research, 40, pages 1489–1506, 2000.

Jacob, R.J.K. & Karn, K.S., *Eye tracking in human-computer interaction and usability research: Ready to deliver the promises (section commentary)*. The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research, ed. by J. Hyönä, R. Radach, and H. Deubel, pp. 573-605, Amsterdam, Elsevier Science, 2003.

Nyström, M., Novak, M., & Holmqvist, K. (2004) *A novel approach to image coding using off-line foveation controlled by multiple eye-tracking measurement*.

Osberger, W., Maeder, A.J. *Automatic identification of perceptually important regions in an image.* Pattern Recognition, 1998. Proceedings. Fourteenth International Conference

Privitera, C. M., & Stark, L. W. (1998) *Evaluating image processing algorithms that predict regions of interest.* Pattern recognition letters 19 (1998) 1037-1043.

Privitera, C. M., & Stark, L. W. (2000) *Algorithms for Defining Visual Regions-of-Interest: Comparison with Eye Fixations*. IEEE transactions on pattern analysis and machine intelligence. Vol. 22, No 9, September 2000.

Reinagel, P., & Zador, A. (1999) *Natural scene statistics at the centre of gaze*. Network: Comput. Neural Syst. 10 (1999) 1-10.

Schneiderman, H. (2000) *A Statistical Approach to 3D Object Detection Applied to Faces and Cars.* PhD Thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, U.S.

Trucko, E., Viel, F. & Roberto, V. *Near-recursive optical flow from disturbance fields.* BMVC 2002.

Wang, C., & Bovik, A. C. (2001) *Embedded Foveation Image Coding*. IEEE Transactions on Image Processing. Vol 10, No 10.

Wolfe, J. M. (1998). *Visual search*. In H. Pashler (Ed.), Attention. London: University College London Press.

Yarbus, A. L. (1967). *Eye Movements and Vision*. New York: Plenum Press.