Charles University in Prague
Faculty of Mathematics and Physics

Diploma Thesis

Jan Sýkora

Methods of digital image processing in non-photo-realistic imaging

Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic

Diploma work supervisor: **RNDr. Zitová Barbara, Ph.D.**

Study program: **Computer Science, Software systems**
Study plan: **Computer graphics**

Rád bych poděkoval vedoucí diplomové práce Barbaře Zitové za neskutečnou trpělivost a pomoc při psaní této práce...

Prohlašuji, že jsem svou diplomovou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 19. dubna 2007 Jan Sýkora

Contents

Abstract

Název práce: Techniky automatického zpracování obrazu v nefotorealistickém zobrazování
Autor: Jan Sýkora
Katedra (ústav): Ústav teorie informace a automatizace, AV ČR
Vedoucí diplomové práce: RNDr. Zitová Barbara, Ph.D.
e-mail vedoucího: zitova@utia.cas.cz
Abstrakt: Navrhovaná metoda provádí cílenou abstrakci obrázku. Uživatel za pomoci počítačové myši určí v obrázku významná místa. Tato jsou při později využita metodami zpracování obrazu k dosažení kýženého výsledku - obrázku, jehož obsah je prioritizován. Prioritizace, jakožto forma transformace obsahu, navádí mimovolně zrakovou pozornost diváka předem definovaným způsobem. Metoda patří do kategorie nefotorealistického zobrazování, neboť splňuje jeden ze základních aspektů - vytváří obrázek, jehož cílem je zkvalitnit přenos informace
Klíčová slova: nefotorealistické zobrazování (NPR), cílená abstrakce, počítačová myš

Title: Methods of digital image processing in non-photorealistic imaging
Author: Jan Sýkora
Department: Institute of Information Theory and Automation, AS CR
Supervisor: RNDr. Zitová Barbara, Ph.D.
Supervisor's e-mail address: zitova@utia.cas.cz
Abstract: We introduce the method that performs meaningful abstraction of an image. By moving the mouse the over the image, the user identifies places he/she finds important. This selection is later applied to various operations of image processing, resulting in an image that expresses the intended prioritization. When the resulting image is later viewed, the structure of the image drives the viewers attention in a predictable way—effect of the meaningful abstraction. The method belongs to Non-Photorealistic Rendering (NPR) as it conforms to the main goal and creates an image that communicate more effectively.
Key words: Non-Photorealistic Rendering (NPR), meaningful abstraction, mouse interaction

Chapter 1

Introduction

We would like to address the creation of Non-Photorealistic Rendering (NPR) of images. More specifically, we will attempt to shed light on the ways in which this creative process of image processing, as subjectively mediated by human interaction, may be used to achieve the desired NPR result. A great deal of theoretical and conceptual research necessarily precedes this work, and we will go on to outline a few such research projects—those of greatest relevance to my own—later in this thesis. We would like to state now, that the proposed method, as presented in this thesis, should not be seen as an attempt to refute or dismiss these prior findings. In the work we will review in the following section, the use eye-tracking technology is employed in order to produce an image which is simplified and thus easier to understand, while at the same time, the image still retains all the desired information. In other words, the produced image is paradoxically both more accurate (through meaningful abstraction) and simplified. The result is a visually less detailed (hence non-photo-realistic), but nonetheless expressive, image.

In this thesis, we will review this technology and the theories which anchor it. Then, in subsequent sections, through drawing upon these and our work, we will show that it is possible to achieve a similarly enhanced NPR result, without the necessary use of, or access to, eye-tracking technology. Thus, one of the primary assertions resulting from this thesis is that such an information-enhanced NPR result can be achieved "at home". By using a different kind of interaction with the system, the results might also be more calculated. The desired output from the interaction is set of the important places at the image. We will study the ways, how the source image might be changed in order to express the importance and

deliver it later to the viewer.

In Chapter 2 we describe the methods and theory that will be later used in proposed method (Chapter 3). Firstly, in Section 2.1 we introduce NPR as an area of *Computer Graphics*. We illustrate the main areas of NPR by examples of methods. Then in Section 2.2 we set a classification that separates the methods based on the communication goal. After in Section 2.3 we employ the user to interact with the system to supply creativity to the processes of computer imaging. Later in Section 2.4 we describe the key principles of image abstraction, as a key tool in feature prioritization.

In the Chapter 3 the proposed method is introduced. We start in Section 3.1 with the description of how the image abstraction is performed. The suggested ways of human interaction and its processing is discussed in Section 3.2. Section 3.3 then, using the previous results, generates the final image, that satisfies the desired communication goals.

Chapter 4 presents the detail of the proposed method implementation, followed by Chapter 5 that contains the conclusion and presents a result.

Chapter 2

Background

In the following chapter I present the ideas and sources that will be referenced further in this thesis.

Firstly, Non-Photorealistic Rendering (NPR), as a field of computer imaging is introduced. One of the most noticeable aspects that NPR has in comparison with Photorealistic Rendering (PR) is creativity. Because creativity is provided by humans, the process of interaction enters the scene. When creativity is combined with the process of image abstraction, the results, which have noticeable influence on the way they are understood by the viewer, are obtained. I'll demonstrate this method in the processing case.

2.1 Non-Photorealistic Rendering

Computer graphics, as a sub-field of computer science, has during its existence incorporated various groups of research. The basic principles of image digitalization and/or representation are described in *computer imaging* and *image processing* performs operations and analysis over represented data. The images might also be entirely generated by computer, in such case the corresponding research falls into the *Computer Generated Imagery (CGI)* category. Needles to say that the research in one group is usually dependant on/used by the methods from the other categories. Anyway, all the methods share the common focus—an image. In most cases this image will, sooner or later, be presented to a viewer. Then it would be evaluated in terms of how it conveys the encoded information. In the other words, how the information is depicted. For the purposes of this work, we distin-

(a) (b)

Figure 2.1: Example of Non-Photorealistically Rendereded images: Picture (a) displays sketch drawing generated directly from 3D model. Picture (b) presents a technique for the digital facial engraving.

guish between two types of image depiction. Photorealistic Rendering (PR) and Non-Photorealistic Rendering (NPR).

The first category denotes a group of techniques that share the goal of realistic images production. As the goal of these methods is clearly set and the evaluation of output image quality (in terms of appearance) is easy, a wide cooperation platform has been established, leading to the development of advanced techniques including ray-tracing, model shadings or numerous illumination models. Output images look more and more as if they were photographs. Currently there are many areas of applied computer imagery, where realism is a key criterion. These include, for instance, different kinds of simulators (plane, army), computer generated movie scenes, etc.

As opposed to PR, which tries to approach reality as closely as possible, the NPR principle asserts that in the field of communication, realism is not important. In other words NPR is allowed to distorts reality, while targeting a given communication goal. Figure 2.1 shows some examples of images belonging to NPR.

As its name suggests, NPR does not attempt to present the viewer with every detail of an image, unlike photorealistic images. In NPR, the ultimate goal is effective, meaningful communication. The priority here is shifted from realism to abstraction. Through the creative process of abstraction, the image may be manipulated to form a bridge of communication between its creator and the viewer;

in other words, NPR may serve as a vehicle for a different, and in some cases more effective, mode of conveying visual information. It is perhaps useful to further clarify the idea of communication. Both PR and NPR naturally, are in the business of communicating to the viewer through images. There is a significant distinction, however, between the two. When an image is produced to be as photo-realistic as possible, its creator assumes that given its faithfulness to reality, the effective communication of the information therein depicted, is guaranteed. In NPR on the other hand, given the fact that the image is not realistic, its creator faces potential misunderstanding between the final image and its viewer. The schema in Figure 2.2 demonstrates the process of information transfer where the depiction is controlled by PR and NPR. As the PR process fully concentrates on the image realistic look (in this case corresponding to arrow (A), relying on the fact that understanding (B) (information decoding) of the result is straightforward. NPR, on the other hand, plays with both processing phases. It allows the creator of the idea to use different and even more efficient ways of expressing ideas, that would be impossible using the ordinary PR process.

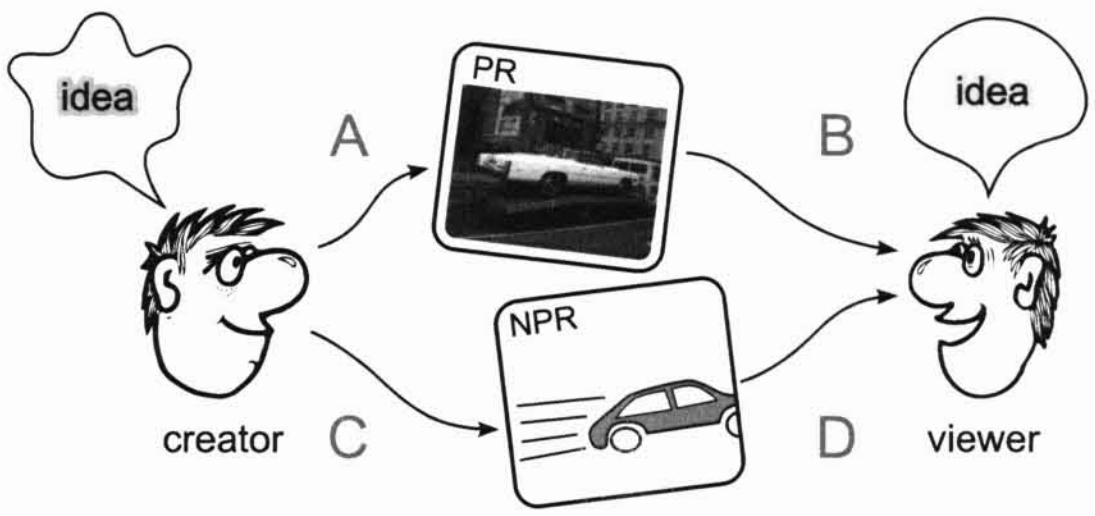


Figure 2.2: Schema of the PR/NPR transfer of information. Photorealistic Rendering (PR) relies on the viewer's visual experience, not concentrating on process B, while putting extensive effort on information encoding A. Non-Photorealistic Rendering (NPR) focuses more on the cognition process D, trying to make communication more effective.

Needless to say, NPR is not something that has been newly invented. It is rather a collection of existing methods and approaches that share a common attribute, i.e. the result produced is no longer evaluated in terms of appearance, but rather by its ability to communicate. From this point of view it is obvious that there is no sharp distinction between PR and NPR methods. It is very common for one method to fall under both categories, but the difference comes with the

context in which the method is used.

For classification purposes, Gooch and Gooch [1] define three general categories into which they split contemporary NPR research. These are *Artistic media simulation*, *assisting the user in the artistic process* and *automatic systems*. In the following subsections we will explain which methods belong to each category and we will show several illustrative examples.

2.1.1 Artistic Media Simulation

> *In all presented cases, the researches did more than simulate the physics of the medium. They also provide high-level tools for both artists and non-artists reducing the time and the skills needed to produce a work of art. Especially those tools that empower the non-artist seem to be a great interest to the graphics community.* [1]

When simulating artistic media in a computer drawing process, we want the result to closely resemble reality. By studying the physical properties of the physical media, the basic principles might be inferred resulting in simulated drawing that produces appealing images. Extensive research in this area allows present-day systems to be capable of simulating not only various media, but even different kinds of canvas materials, drawing conditions, etc. Figure 2.3 shows an example of different techniques as created by the program ArtRage[1].

Example

As an illustrative example of the research in the field of Artistic Media Simulation we describe the simulation of oil-paint (on Figure 2.3, first from the top).

To accomplish such an appealing look, several factors had to be considered and evaluated. Gooch & Gooch [1, p 31-49] separate the simulated painting process into the tasks. The first is stroke placement.

The strokes are placed either by the user or automatically. When placed manually, the limitations of user input force the process to determine correct values for brush orientation or its pressure based on available data. The obvious way is to let the user set the pressure with additional control and compute orientation with respect to the tangent of the followed path. With automated systems it is quite different. In both cases, the direction of the brush corresponds to the direction in

[1]see `http://www.ambientdesign.com/`

Figure 2.3: Example of oil paint, pencil, chalk, marker and crayon

which the stroke has been painted. The image is re-stylized using the brush while the stroke and brush orientation are then determined in such way that the edges are not crossed, but rather that their contour is followed.

The Figure 2.4 shows more examples of oil painting. The arrow illustrates the direction in which the stroke has been made. The resulting look of the stroke depends (according to [1]), on various factors:

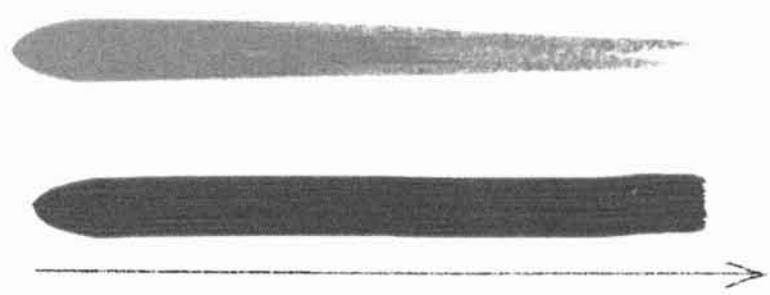


Figure 2.4: Example of brush stroke on a paper canvas

1. **Brush**
   When painting with a hairy-brush the hair contains paint and is responsible for its transfer onto the canvas. The amount and resulting structure of paint transferred depends on various factors, such as stroke phase (beginning, end, ...), brush pressure, brush orientation, loading (amount of the paint on a brush at the beginning of a stroke; here, the red stroke in Figure 2.4

has lower loading than the blue one. The brush itself might have different shapes, different hair configuration (toughness, absorbtion), etc., making an the exact simulation of the paint process very complicated.

2. **Substrate**
   The substrate influences a painting in several ways. The structure of the canvas might serve as a barrier disallowing the paint to smoothly spread, becoming visible when the brush is getting out of paint. In Figure 2.4 it is the right part of the red stroke that - due to lack of paint - unveils the structure of the canvas (rough paper). The material of the substrate might have a direct influence on the spreading of the paint and its placement. For the simulation of oil paint, the program tracks the actual amount of water for everywhere. When the stroke is applied on a highly absorbable canvas, the paint dries out very fast disallowing mixing of the colors. Another important factor—which is closely connected to the previous one—is color diffusion. When painting with watercolors on a very diffusive canvas, not only will the color dry very fast, it will spread in space and mix with the neighboring colors.

3. **Media**
   This describes how the physical properties of the media apply with the brush. It defines the ways in which the color is applied onto the canvas, how it is mixed with other media already applied (not necessarily media of the same type), depending on the brush pressure and angle how original color will be distributed along the path (look at the colors of the blue stroke in Figure 2.4)

Now we are able to simulate different kinds of artistic media using a computer. Nevertheless, this is just the way output should be generated to have the desired appearance. We are able to show brush strokes or lines drawn with crayon or pencil, but what we are missing is the information of what to draw. This is where the user enters the scene.

2.1.2 Assisting a User in the Artistic Process

User-assisted image creation contains methods that literally allow any user to become an artist. The main idea is to incorporate the skills of human artists into an expert system and the skills are later guiding the user in a drawing process. This might in the end result in images with a hand-crafted look and feel. [1].

The methods in this category help the user to do something he/she cannot do without help. By studying the actual drawing processes, important rules might be inferred and applied later, guided by the human interaction. We might say that the quality of communication between the user and a computer is enhanced thanks to the process, since the user needs less effort to achieve the desired result.

Example

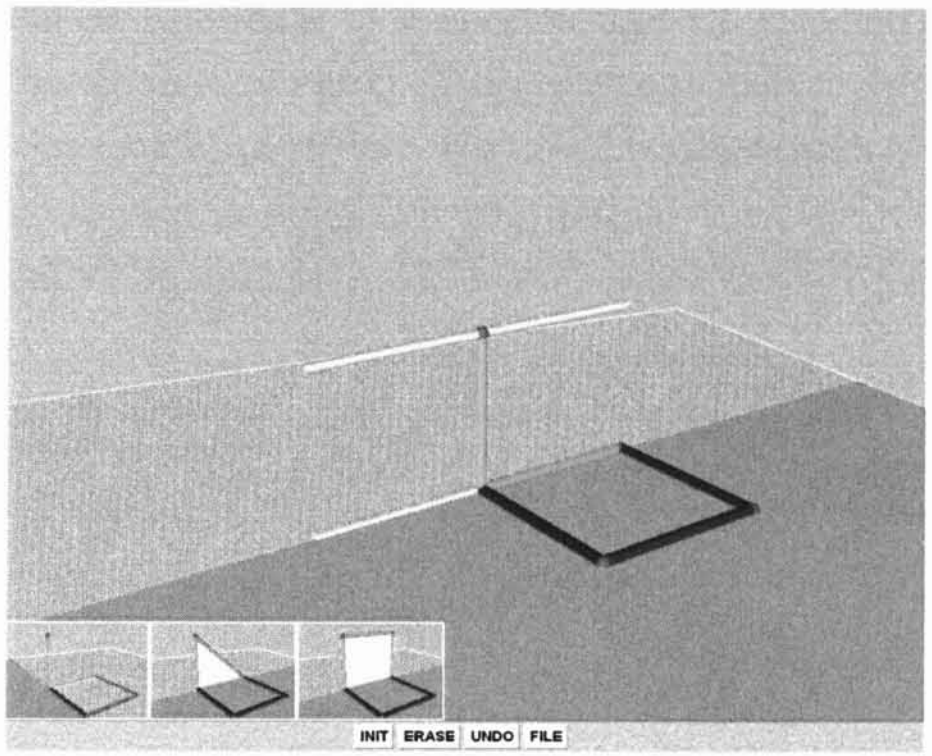


Figure 2.5: Chateau - the suggestive interface for 3D drawing

Figure 2.5 shows a suggestive interactive interface for the 3D drawing called `Chateau` [2]. The effectivity of communication is enhanced by suggestions, which the software makes based on the estimate of what kind of operations the user might want to do next. To determine this, several principles - like symmetry, recurrence and similarity - are used. If we look at Figure 2.6, there is a frame on the ground with a single beam pointing up. The program makes several suggestions. First, drawn in magenta, it offers creation of new beams, which are symmetrical to the existing ones. Secondly, it offers (in the lower-left corner) the creation of a new face. And finally, it offers a change of the working plane which is to be perpendicular to the existing one while rotated around the upward beam. Similar suggestive principles might also be found in `SketchUp` software [3] from Google, which is a tool for easy 3D modeling.

[2] see `http://www-ui.is.s.u-tokyo.ac.jp/~takeo/research/chateau/chateau.htm`
[3] see `http://www.sketchup.com/`

2.1.3 Automatic Systems

> *The automatic image creation research focuses on system, that are able to produce artistic images with a previously defined communication goal.*[1]

Automatic Systems deliver NPR results without any user interaction. Because of this fact, the method belonging to this group are acknowledged in areas outside the computer science research. For example NPR processed X-ray image might emphasize the potential problems of a specific nature (fracture, tumor, heart attack, ...) lowering the chances of error. This category also includes method capable of simulating different painting styles. Figure 2.6 illustrates the result of the automatic transformation of a photograph to a painting which looks like it has been painted pointillistically[4].

Figure 2.6: An automatically created picture in pointillism style

In the previous section, NPR research was grouped into three categories. For the purposes of this work, however, we have established a different categorization. This is because we construe communication with the viewer (D in Figure 2.2) to be the main focus of NPR, which the classification should follow. Detailed description and differences between the categories are described in the following section.

[4]created in `Virtual Painter`, see `http://www.virtualpainter5.com/`

2.2 Communication-Oriented Classification

The suggested classification judges the research based on the effect the image will have on viewers. More specifically, this is the primary goal of the communication that occurs as the viewer tries to decodes and understand the picture (corresponds to the arrow D of the Figure 2.2). There are two groups defined - *Impressive imaging* and *Communicative imaging*.

2.2.1 Impressive Imaging

Figure 2.7: An example of an oil-paint simulated image

So-called Impressive Imaging produces images that stick to a given form, i.e. they look a certain way. The resulting pictures are interpreted subjectively, mainly based on individual experience and taste. As those vary much among people, the concrete interpretation is *unpredictable*.

As an example we present a method that simulates an oil-painting. An example result generated using this method is presented in Figure 2.7. The source image, as shown in Appendix A, is a photograph, which is - necessarily - photorealistic. After the process was applied, the form changed and the image became non-photorealistic. Furthermore, it is an impressive non-photorealistic image, because I, its creator, chose the form that best expresses the informational content I

want the viewer to gain: I felt this form evoked the appropriate mood and atmosphere. In this case the form of the image was the focal point of the modification process.

For Impressive Images, the target form is the only thing that needs to be chosen before processing. The informational content of the image (what is displayed) is not manipulated in terms of meaning and the only modification involves restyling with respect to chosen form.

In summary, in Impressive Imaging, that form is chosen which expresses best the informational content that the creator of the image wishes to convey to the viewer. We can thus say that in Impressive Imaging, the form is what is applied to the content, targeted as the primary vehicle of meaning. The *form of the image* is at the center of Impressive Imaging.

2.2.2 Communicative Imaging

Figure 2.8: An example of a processed image from the Communicative Imaging category

So-called Communicative Imaging produces images that convey certain information, to which the form is secondary.

For this purpose the information content of an image is subject to intentional manipulation. It can be said that the information undergoes a process of *prioritization*. This means that through various processes (to be described later in

Section 2.3), the creator of an image defines the importance of the image aspects. This will be later used in the processing, to determine the amount of expressivity the particular image feature should convey. Therefore, since the information content of the image has undergone prioritization, not all information in the image must be equally depicted. That which is of less importance may be relegated to the background or even omitted entirely. At the same time, it is equally possible to emphasize information which is of higher priority. So, it is possible to raise or lower detail levels. As a direct consequence of either of these operations, the levels of detail can be manipulated with respect to each other, resulting in *visual expression of the original prioritization.* The final image is thus constructed in such a way that it is easier for the viewer to interpret the image as was intended by its creator [4]. This kind of imaging is eminently useful as a device for communicating specific information. Form here becomes secondary to communicative value. Figure 2.8 shows an example of image belonging to this group. Emphasized portion of the image content drags immediately the user attention, which was the primary goal of the prioritization.

In communicative imaging, prioritization is intentional and is supplied by the user. The NPR process then transforms the image into the result. As we mentioned previously, the information is encoded independently of the actual depiction form used. When we go back to the intentional and creative aspect of the prioritization, user is responsible for supplying the required input; as we will discuss further, this is a process of human interaction.

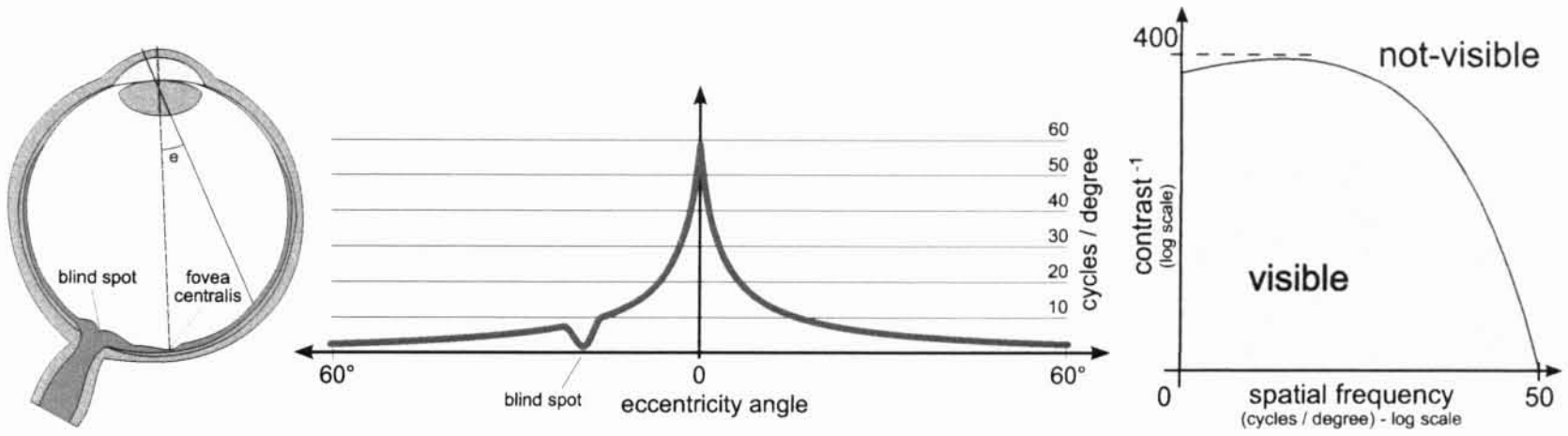
2.3 Human Interaction

For the meaningful abstraction, the creativity is the primary source for defining the salient areas of the image. Santella and DeCarlo had evaluated in [4] the results of automatic and user assisted methods in terms of the salient places selection. The results had confirmed, that meaningful abstraction is something computers can not do as of yet, as deeper understanding of the image structures is required. The creativity is supplied through the process of human interaction.

In our case, the human interaction is provided by the mouse motion. We will now describe processing of this interaction in detail. Lets us imagine we have an input image and the creator has an idea according to which he wants to have the picture processed. The idea is first supplied as input to the chosen NPR throughout the interaction process. The method then processes the input image with respect to the idea, resulting in an image whose visual form expresses the idea. When

such an image is later viewed, the depiction form drives the viewer to understand the idea. When the viewer decodes the idea, it can be stated that communication has occurred and the information has been transferred.

2.3.1 Visual Perception



(a) (b)

Figure 2.9: (a) shows the schema of the human eye along with a graph of maximal perceivable frequency. The perceivable frequency is dependant on the eccentricity angle e (in degrees). (b) Graph of a maximum perceivable contrast depending on spatial frequency

The main focus of NPR is communication with the viewer. As we focus on images generated by NPR processes, the communication that occurs is mainly of visual nature. This means that the NPR results are perceived by the visual sense - with the eye. Therefore human visual perception has to be further examined to form a background for the processing.

Even though human vision and cognition are both very complicated processes, thanks to recent studies in the psychophysical field [2] some conclusions may be drawn. When talking about unchanging imagery, several experiments showed that

- non-uniform density of photoreceptors, which is far highest in the center of the retina (fovea centralis) forces the vision to look precisely at each fine detail separately in order to register it.

- we identify two main types of eye motion (relevant to this work) - rapid motion (saccade) and stabilizing motion (fixation). Overwhelming majority of visual processing takes place during fixations, and saccades are performed mainly to position the eye to look at the right spot.

- there is a relation between the length of the fixation and the amount of visual processing. Exploratory fixations tend to be short.

There have been several experiments to define what the human eye — as a sensory organ — is physically capable of registering. One of the most fundamental tried to determine maximal perceivable frequency at maximum contrast (depicted as repeating sinusoidal gradings). From the results, the perception model of human eye was determined, defining the limitation of the human eye as:

$$G = 60.0\ cycles/degree \tag{2.1}$$

$$M(e) = \frac{1}{1 + 0.29e + 0.000012e^3} \tag{2.2}$$

$$H(e) = G \times M(e) \tag{2.3}$$

where G stands for the maximum perceivable frequency at maximum contrast in the most sensitive part of the retina (fovea centralis). The $M(e)$ is a falloff factor for the given eccentricity angle e, which is 1 at the center of the fovea and decreases as the angle e increases. $H(e)$ then gives us the overall visual acuity for a given angle e. The visual acuity rapidly decreases with for higher angles, as illustrated on graph in Figure 2.9 (a).

The second experiment we mention tried to identify distinguishable contrast for a given frequency (measured on the perception of the blurry stripes) [3]. The inferred equation has the following form:

$$A(f) = 1040(0.0192 + 0.144f)e^{-(0.144f)^{1.1}} \tag{2.4}$$

where the f corresponds to the examined frequency resulting in a limiting inverse contrast value A. To determine the contrast value c, we use the Michelson contrast, that is defined as $\frac{l_1 - l_2}{l_1 + l_2}$, where l_1 and l_2 are the corresponding intensities. Picture (b) of Figure 2.9 displays a graph of recognizable and non-recognizable contrast. Having established the foundation for the evaluation of visual perception, we now present one type of Human Interaction - the process of eye-tracking. Also the process of visual perception in time will be examined.

Eye Movement

> *Eye movements over imagery are directed in a meaningful and economical manner, and are tightly linked to cognition.* [4]

Human vision and cognition are very successful and efficient in scanning the world around. There is relation between positions where the user looks and the subjective semantic informativeness of the image [2]. We will use this information in the process of prioritization (see Section 2.2) to define places the viewer finds important.

As suggested by Santella and DeCarlo in [2] we might directly obtain the positions where the user is looking, by using a special device called an *eye-tracker*. When properly calibrated, such a device is able to record and convert the rotation of a human eye into the Point Of Interest (POI).

The model introduced in 2.3 corresponds to the physical limitations of the human eye. Now we focus on visual cognition in time, when the visual focus changes. We have to note, that we are talking about the static imagery, where the observed image itself is not changing.

As stated for the $M(e)$, perception quality is by far highest in the center of vision. This implies that people are capable of detailed examination (with cognitive processing in the background) only of a small central part of the image they see. So, in order to examine larger areas within the available visual field, the eye has to move (rotate).

As mentioned earlier, we recognize two main types of eye motion - rapid motion called saccades and stabilizing motions called fixations. The saccade is used when the vision is in the process of changing its focus. The motion speed usually exceeds 10 m/s and during saccades there is not much visual processing performed. In other words, we might omit the saccades in the prioritization process, as the vision cognitive processes are minimal.

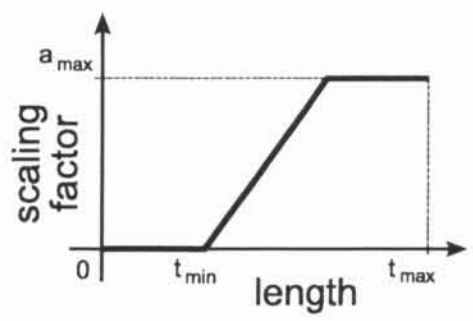


Figure 2.10: attention model

The overwhelming majority of visual processing takes place during fixations. Santella and DeCarlo [2] had set the relation between the the amount of processing and the fixation length as a piecewise linear function. We may call this a *simple attention model* and Figure 2.10 show a graph of this function.

Setting the relationship between the amount of processing and the subjective local importance, the human interaction can be used to determine the content *prioritization*. As the the creativity is supplied through the human interaction, meaningful abstraction can be achieved.

2.4 Image Abstraction

Santella and DeCarlo state in [3] abstracted image directs the viewers attention to most meaningful places. This allows the viewer to understand structures of the image without conscious effort. When such abstraction is done intentionally with previously defined goal, the viewers cognition might be influenced in a desired way. This corresponds to the *prioritization* as required by Communicative Imaging. Clarification of meaningful structure of an image influences the quality of the information design [3]. The image abstraction might be used within NPR to accomplish the primary goal—communicate more effectively.

All the methods of NPR allow to include additional information in addition to the content depicted. Impressive Imaging applies form to informational content, which results in specific atmosphere of a picture. However, Communicative Imaging modifies the informational content by prioritizing specific aspects of it, while the selection of the resulting form is not important. The main NPR characteristics is, however, valid for both groups and so I would like to quote the description of NPR presented in [1].

> *NPR brings together art and science, concentrating less on the process and more on the communication content of an image. In photo-realistic rendering, it is hard to neglect detail; in fact the highest detail is generally preferred, even if this high level of detail makes the image clustered and confusing. The level of detail on NPR varies between images and can be adapted across a single image to focus the viewer's attention. NPR is now being acknowledged for its ability to communicate serious ideas.*

2.5 Case

Having provided background for NPR research, we now present two methods falling within this group. They both perform meaningful abstraction of given photograph and belong to the category of Communicative Imaging (see 2.2.2). In both presented methods, the level of abstraction is not uniform across the image and the amount of local detail is driven by human interaction, namely by the process of *eye-tracking*. This involves usage of a device called *eye-tracker*, which tracks the position of the pupil and estimates the appropriate eye rotation. From this, the POIs are determined and processed resulting in a set of fixations. These

serve as input for the process of prioritization, which is the key principle behind the meaningful abstraction.

In the first paper [2], the process of painterly rendering [1, 4.3 p.63] was chosen to perform the abstraction of the input photograph. The prioritization is applied during the processing by alternation of the brush parameters with respect to the recorded fixations. The produced image has more detail in areas that the user found interesting—this is caused by the usage of the smaller stroke size in the painterly rendering process. Conversely the selection of the higher brush diameters clears out the detail from the unimportant places.

In the method presented in the second paper [3], the abstraction of the original image is achieved by the image segmentation. This process partitions the image into the adjacent regions of identical color ([10, p.612] [9, p. 353]). As this method processes the image, it removes the fine detail and thus the produced image is abstracted. In order to be able to control the local level of detail, Santella and DeCarlo perform even segmentations of the down-scaled variants of the image.

This results in a set of images with different level of detail, as the down-scale operation strips down the high-frequency information (consequence of the Shannon sampling theorem [11]) that corresponds to the edges acting as the region delimiters. The segmented image with lower resolution thus has less detail than its variant with higher resolution. When all the variants are segmented, the hierarchy tree is build up. This assigns to each segmented area in one step less detailed (down-scaled) image its containing area. Such parent area is chosen that is best with respect to the overlap and color difference factor. Figure 2.11 shows the schema of the spatial hierarchy and the corresponding assignment tree.

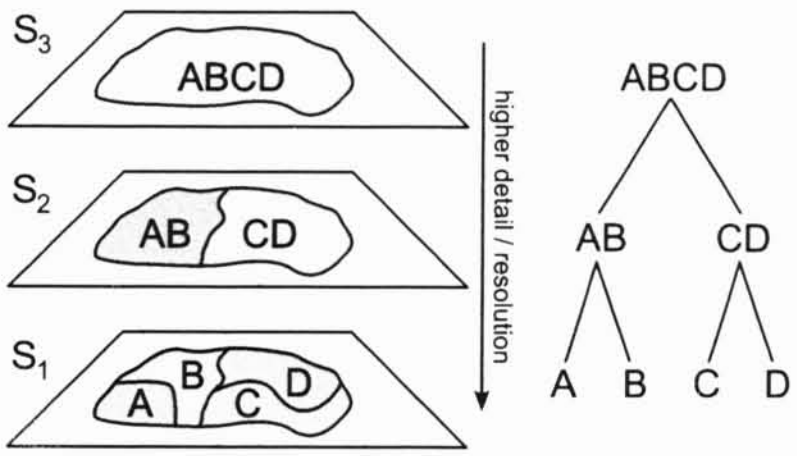


Figure 2.11: Schema of a built hierarchy across the segmented images with different level of detail as used in [3]

The method then processes the image starting with the lowest detail and splits each area into its child area if the child area meets the visibility criteria. This criteria takes into account the region size, contrast relation to its neighbors and visibility with respect to the closest fixation.

Figure 2.12 shows the results generated by the methods presented in [2] and [3]. The content of both processed images has been modified with respect to data

(a) (b) (c)

Figure 2.12: The input photograph (a) has been processed, with respect to the fixations recorded by the eye-tracker, using the painterly rendering (b) [2] and composing the segmentations with different levels of detail (c) [3].

supplied by the eye-tracking device. As evaluated in [4], such abstraction of an image influences the viewers cognition process in desired way. More specifically, the abstracted images convey the encoded information, that is later decoded by the viewer resulting in a more effective communication. The main goal of NPR has been accomplished.

Chapter 3

Proposed Method

The NPR method presented in this thesis performs a *meaningful abstraction* of an input image. Given a photograph and information about the content prioritization, NPR produces an image that emphasizes the desired locations. The method belongs to the category of communicative imaging (see 2.2.2) as it modifies content of an image with respect to a given communication goal.

The basic principles of the method are built on ideas presented by A. Santella and D. DeCarlo in their works (see 2.5).

In the method, the human interaction plays a key role in the meaningful abstraction process. Prioritization of the resulting image is intentional and would comply to the expressive goal of the viewers. This approach naturally delivers far better results than methods for automated identification of the salient places [6] as the human interaction results from experience and has extensive and complicated cognition processes in the background.

Figure 3.1 shows the schema of the proposed method. The box I identifies the image—photograph that is a required input for our method. As the image is processed with respect to the human interaction, the box T represents the track that might be either imported or recorded from a position of an input device (e.g. mouse).

The processing is then divided into two separate parts. First part might be called the preparation phase. The original image is abstracted with different levels of detail and along with the original image packed into the Processing Package (PP) (on the scheme displayed as a green box with dotted line).

The second part starts with the processing of the loaded/recorded track T.

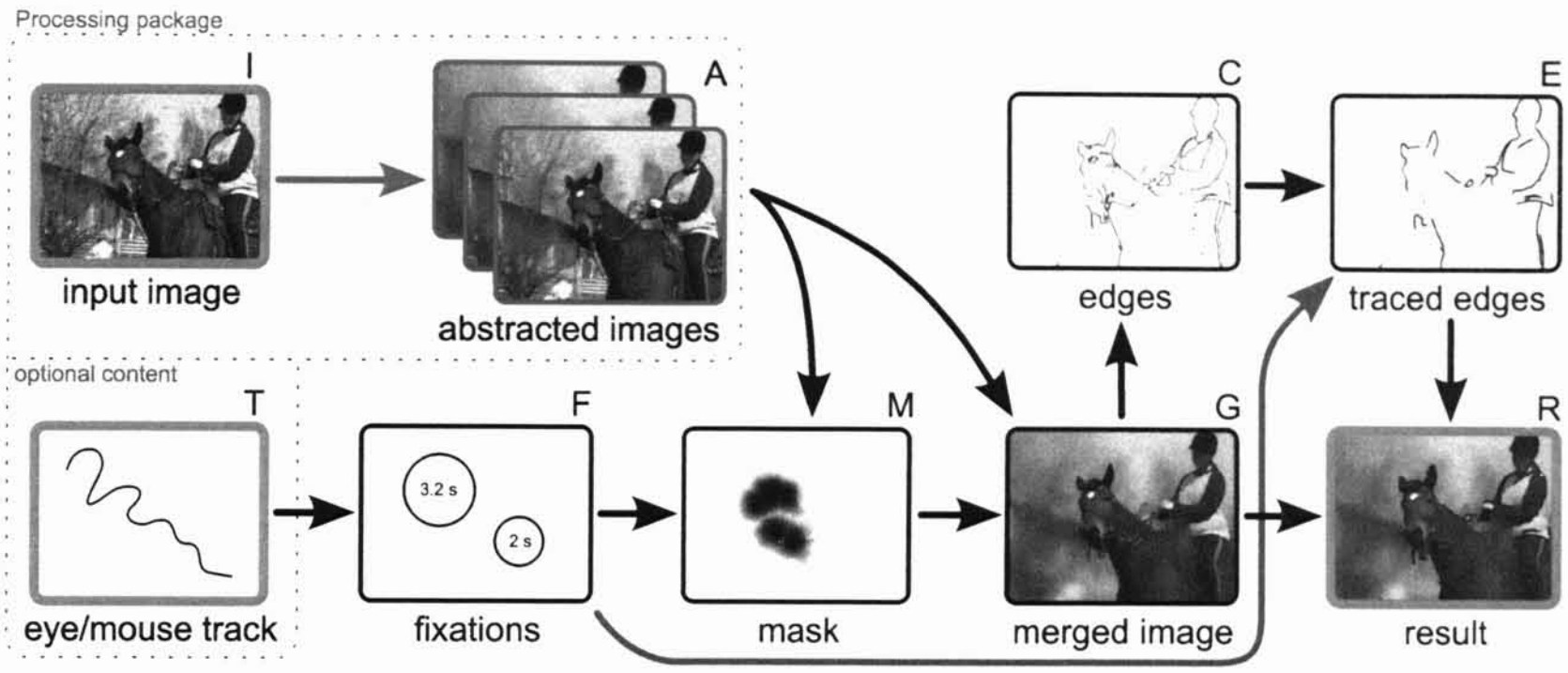


Figure 3.1: Schema of the proposed method, where the arrows denote the processing flow. Having the input image I and the eye/mouse track T on input the method, it produces the resulting image R.

This is filtered and clustered, resulting in a set of fixations F. The fixations are then, along with the set of the abstracted images A, passed to the mask processor and the *detail level mask* M is generated. This mask reflects the desired local detail and is generated with respect to the recorded interaction and the image. By applying the importance mask on the set of abstracted images A the merged image G is generated. This image has non-uniform level because the content prioritization process took place. Afterwards, the edges are extracted from G using the Canny edge detection algorithm [8] resulting in a binary edge image C. This is processed with algorithm of edge tracing and the vector representation of the edges is generated. After filtering out the non-satisfying edges, the local edge width is determined with respect to the fixations configuration F. The edges are then rendered in an appropriate color into an image E. When this image is overlayed over the merged image G, we get the resulting image R.

3.1 Image Abstraction

The process of *prioritization* is in our work implemented so that the resulting image is a combination of several images with different level of detail. As stated in [4], an abstracted image with non-uniform levels of detail drives the viewer's attention to the places with high detail—the content is prioritized. The processed images are after the processing assembled, forming a Processing Package (PP),

where every abstracted image has assigned level of relative detail. This number is 0 for image with low detail rising with increasing detail to the maximum value of 1. The examples presented in this thesis are generated and using three levels of abstraction—detail set to 0, 0.5 and 1. For abstracting the source image our method relies on algorithm of denoising. When we state that the unimportant edges are correspond to the noise added to the image, the process of denoising removes them. As in the resulting image, there will be less edges with smooth areas in between, we can say, that the image is abstracted as the fine detail has been removed. The proposed method requires on its input a set of the abstracted images with different level of detail. As in our case the level of abstraction corresponds to the amount of the edges, we have to be able to control this amount in processing. Denoising based on minimization of the Mumford-Shah functional seems to be the right choice.

3.1.1 Denoising

The denoising is a process that tries to identify and remove the noise from a given signal. By noise we mean random errors that might occur during the capture, transmission or processing [8]. As the problem is ill-posed with respect to the noise and the original image, some additional conditions have to be set. First, we define the basic notation.

Let have the picture as a bounded color function $u : \Omega \to [0,1]^n$ where n is number of channels. $a = (x, y)$ denotes the location in Ω, $|a| = \sqrt{x^2 + y^2}$ denotes Euclidian norm. $|E|$ would be Lebesque measure of $E \subseteq \mathbb{R}^2$ which might be considered to be equal to the area of E.

The problem may be formulated then as finding the solution of

$$z = u + n \tag{3.1}$$

where z is the input image, u denotes the original image to be recovered and n is a signal independent noise.

When we say that the noise in the result image is responsible for the fine detail, by removing it the image becomes by definition abstracted. We assume n to have constant variance of δ^2 and zero mean value.

$$\int_{\Omega} (u - z)^2 \, dx = |\Omega|\delta^2 \tag{3.2}$$

$$\int_{\Omega} (z - u) \, dx = 0 \tag{3.3}$$

Let $Q(u)$ denote some regularization functional of the estimated original image. The problem can be then formulated as finding $min_u Q(u)$.

The implementation of finding the minimum is an iterative process. Every step of the iteration lowers the $Q(u)$ as compared to the previous. The value decreases significantly just for few initial steps, and as the experiments have shown, performing ten steps is enough for both the the result quality and the processing time.

The suggested functional is based on the Mumford and Shah complex energy function [5] designed especially for the image segmentation. In addition to the u, it adds another variable—the discontinuity set $S, S \in \Omega$. More specifically, S is one dimensional set where u is not continuous. The functional has then following form

$$Q_{MS}(u, S) = \int_{\Omega} |\nabla u|^2 + \mu \mathsf{H}^1(S) \tag{3.4}$$

where H^1 denotes the 1-D Hausdorff measure. The gradient ∇u is defined everywhere outside S.

From the [5] the regularization term Q_u would then have following form

$$Q_{MS}(u, K) = \|u - z\|^2 + \lambda \int_{\Omega - K} |\nabla u|^2 + \mu \int_K a \tag{3.5}$$

During the minimization, the first term $\|u - z\|^2$ ensures the resulting image would not be completely different than the original one. K is the discontinuity set and the parameters λ and μ control the process. By altering the last two, results with different appearance might be achieved. As $\int_K a$ corresponds to the overall length of the edges, the parameter μ controls the amount of edges and thus the amount of detail. By experimenting with different values we found an optimal combination of parameters that produce the abstracted images with an appropriate level of detail. Those are shown in Table 3.1.

Since we are dealing with color images, the method has to be further extended. The process of minimization of Q_{MS} is performed on each image channel separately. Some other possibilities were evaluated, such as prior conversion to different color

detail level	λ	μ
low	0.005	0.002
medium	0.01	0.001
high	0.02	0.0005

Table 3.1: Level of detail and appropriate processing parameters.

(a) (b) (c)

Figure 3.2: The original image (a) has been processed with the process of denoising, removing the noise (b) resulting in an image (c).

space (YHS). However, the experiments have shown that even when the RGB color space is used in processing, the resulting quality of image abstraction is good. Figure 3.2 showed an example of denoising of the photograph (a). The image (b) corresponds to the noise identified and removed from the photograph (a), resulting in an abstracted image (c).

By altering the control parameters λ and μ, the abstracted image with different levels of detail is generated. By default there are three levels of detail generated—low, medium and high with d_i set to 0, 0.5 and 1. The exact values of the control parameters are presented in Table 3.1. The generated images are supplied along with the original image in PP to the processing application. This determines the content prioritization by the user interaction and performs the meaningful abstraction generating the image with a given communication goal.

3.2 Human Interaction

The human interaction required for the processing involves the tracking of the mouse cursor and import of the prerecorded data.

In mouse tracking, the image is shown on screen and user is instructed to move the mouse cursor over the places he/she finds important. Meanwhile, the exact

position of the mouse is recorded, supplying in the end the mouse track. The experiments have shown that such recorded track follows the principles of Visual Perception (see 2.3.1). We think it is because when the viewer is instructed to identify important places, he or she moves the mouse cursor over and, guided by the visual experience, keeps the mouse steady for the time that is close to the time needed to visually process the location. One of the main advances of manual identification of the important places is that the process is intentional and may be planned before recorded.

As stated in [4], there is a relation between the POIs and the informative content of the image. Having the recorded track t, based on the 2.3.1, we are able to determine the areas of interest—fixations. The track may be defined as

$$t = \{p_i, p_i = (x_i, y_i, start_i, end_i) \mid 0 \leq i \leq n\} \tag{3.6}$$

where (x_i, y_i) is position in a screen space and $(start_i, end_i)$ denotes the time the attention dwelt on a particular place and n is the number of samples. The picture (a) of the Figure 3.3 shows the recorded track points as series of black circles, where the red line connecting them shows the order in which whey have been recorded.

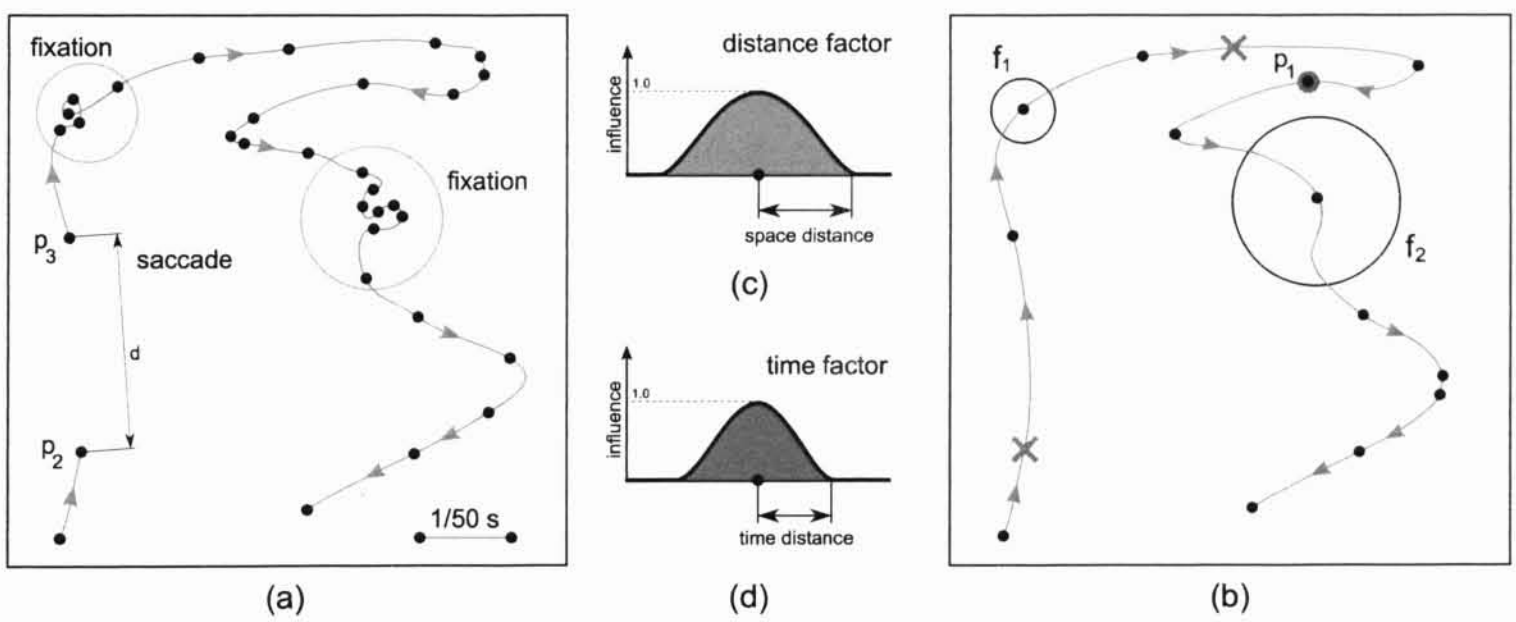


Figure 3.3: Processing of the recorded track (a) with respect to the distance factor (c) and time factor (d) results in the set of fixations (b).

Having the recorded track, we have to process it in order to get the information about which places the user identified as important.

3.2.1 Processing

The track t as defined in 3.2 consist of series of samples with positions $pos_i = (x_i, y_i)$ and time range $(start_i, end_i)$. We further define the length of the sample as $length_i = end_i - start_i$. The first step in the processing is the identification and removal of the saccades. As mentioned in 2.3.1, not much visual processing occurs during saccades and they might thus be ignored in task of identifying places the user finds important. We define the velocity v_i for sample i as $v_i = \frac{d(pos_{i+1}, pos_i)}{lenght_{i+1} - length_i}$ where $d(a, b) = \sqrt{(a_1 - b_1)^2 + ... + (a_n - b_n)^2}$ denotes the Euclidian metric. When looking at the picture (a) of the Figure 3.3, the yellow arrow denotes the distance d between the points p_2 and p_3.

Based on the simple threshold of the velocity value, that is in our case set to the 50 mm/s, we remove the samples with the speed higher than the given limit.

The green crossed points in the picture (b) of the Figure 3.3 identify the samples removed due to high velocity. In the following step, the clustering of the samples is performed. Points that are close to the others in terms of location and time are merged together into a cluster c_i. This is, in further processing, treated as a single sample with length equal to the sum of contained points. The clustering is performed iteratively. For each step the cluster score is computed and, based on this, the clustering is terminated or continues. The score corresponds to the sum of the distances the points have shifted since the last iteration. In each step, a new position of the sample t_i or cluster c_i is computed as a weighted average of close sample positions. The appropriate weight is computed for every sample as a product of the *distance factor*,the *time factor* and *sample/cluster length*. As shown in the picture (c) of the Figure 3.3, the distance factor f_t decreases with the other sample spatial distance, while the time distance is dependant on the difference of the *start* and *end* timestamps. There is maximum space and time distance defined, limiting the influence to the close neighbors. For illustration of the effect of the time factor we can look at the point p_{15} on a picture (b), that has not been merged into the cluster f_2 although it is spatially close enough, as the time difference exceeds the *time distance*.

After the process of clustering finishes, each cluster represents the point where the attention dwelled for longer time. The clusters are now filtered leaving out the ones with short duration—brief fixations (see 2.3.1). Resulting clusters form the set of fixations $f_j = (x_j, y_j, length_j)$. Those will be used in further processing as a vehicle of the content subjective *prioritization*.

3.3 Image Synthesis

The merged image, as symbolized by the block G in the Figure 3.1, is generated from the set of abstracted images a_i and with respect to the fixations f_j. The produced image will have non-uniform level of detail having higher level of depiction for the places with higher detail requested. We generate the *detail level mask* M, which helps us to create the image that expresses the desired prioritization. Using this *detail level mask*, the abstracted images are processed resulting in the merged image. To accentuate the prioritization the edges are extracted and overlayed resulting in a final image.

3.3.1 Detail Level Mask

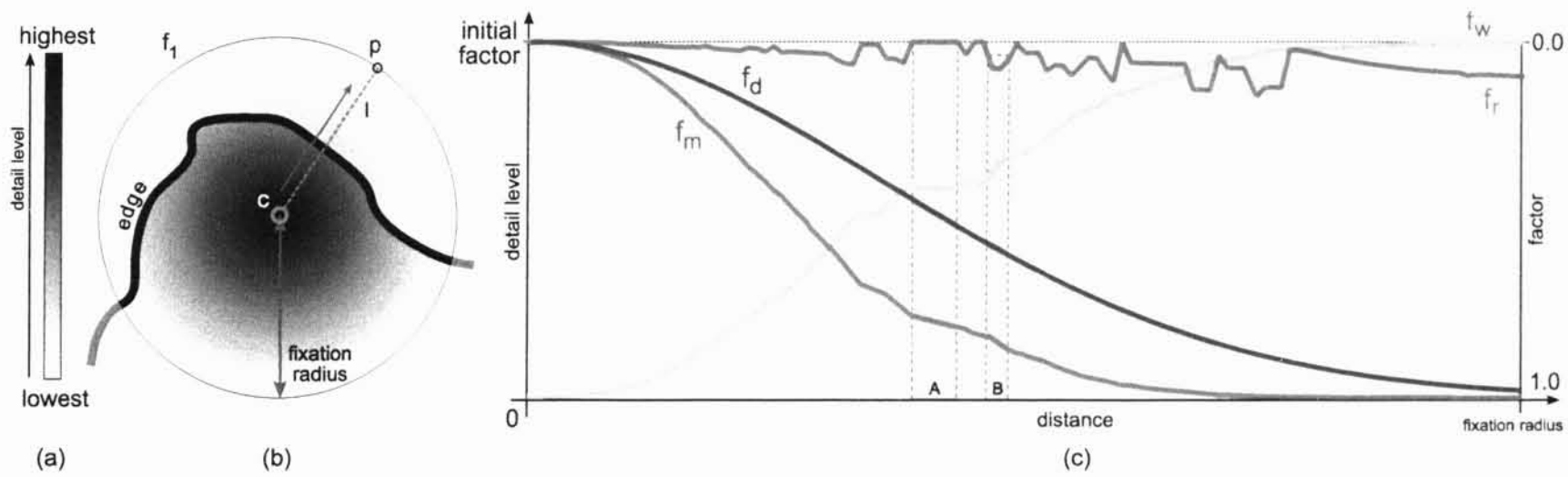


Figure 3.4: Rendering of a fixation (b) to a detail level mask. The graph (c) on the right illustrates the influence of the edges to the resulting detail level

The *detail level mask* is defined as a function $M(x, y) \in \mathbb{R}, 0 \leq M(x, y) \leq 1$ and for every pixel in the image $x, y \in \mathbb{N}, 0 \leq x \leq x_{max}, 0 \leq y \leq y_{max})$ its value correspond to the desired level of local detail. The value of 1 denotes an important pixel, while the value of 0 identifies a pixel that was completely out of the interest. The mask itself is generated from the fixations using the process of *fixation drawing*.

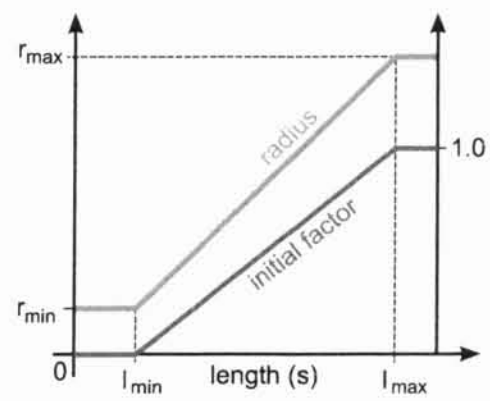


Figure 3.5: Graph of the fixation radius dependency on its length

The process is repeated for every fixation and starts the determination of the fixation influence radius that is based on the fixation length with respect to the perception model (see 2.3.1). The radius defines a circle

centered at fixation position and the fixation might influence the mask M only in this area. Fixations with time lower than l_{min} (in our case set to $1/10$ s) are just brief fixations and does not mean much in terms of content prioritization. Needles to say that the radius r_{min} is usually set to 0 meaning that such fixations do not have any effect at all. As the length of the fixation raises, the influence radius increases maximizing at r_{max} for lengths equal or longer than t_{max} (in our case set to $3/10$ s). The *initial factor* of the fixation that corresponds to the initial detail level (in the fixation center) is also dependent on the fixation length. It is proportional to the length starting at 0 for length $l \leq l_{min}$ increasing to 1 for lengths $l \geq l_{max}$.

Fixation drawing process is illustrated in Figure 3.4. The bar (a) shows the scale of detail of the mask and (b) illustrates the drawing operation of one fixation. For every point p on the fixation influence boundary, the line l is traced from the center of the fixation c (in the direction of the arrow). For every pixel of the line, the value of the mask under is updated. New value corresponds to the maximum of the old value and newly computed detail, ensuring that maximum possible detail for every point is kept. The detail level starts at the *initial factor* decreasing as the line tracing proceeds. The actual value that is drawn to the mask is controlled by several factors—the distance from the center (*distance factor* f_d) and qualities of the crossed edges (*edge withdrawal factor* f_w). The *distance factor* in our case starts at value of 1 decreasing exponentially with equation $f_d = \frac{1}{e^{(2d)^2}}$, where the d is the distance from the fixation center c. The *edge withdrawal factor* f_w is influenced by the edges that are in the process of the line tracing crossed. The influence of the crossed edge is expressed as an *edge influence factor* f_r, which corresponds to the *contrast visibility* (see 2.3.1) ratio of the adjacent traced pixels k_j and k_{j+1} on the traced line l. Let the $c(a, b)$ be the contrast estimation function of two intensities a and b defined as $contrast(a, b) = \frac{a-b}{a+b}$ (corresponds to the Michelson contrast [3]). We have set the color contrast for every pixel to equal to difference of the luminosity as defined in [9]. The *edge influence ratio* f_r is computed as

$$f_r = \frac{a_{max} - contrast(lum(k_j), lum(k_{j+1}))}{a_{max}} \tag{3.7}$$

where a_{max} is the minimal perceivable contrast for current eccentricity angle (measured from the current fixation center) and $lum(p)$ is the luminance of a pixel p. The *edge withdrawal factor* f_w starts at value of 1. In every step, having the value of f_r the f_w is updated as $f_w = f_w * (1 - f_r * (1 - f_d))$. This means that

the value is decreased based not only on the quality of the crossed edge, but also on its distance from the fixation center c. The segment A in the image (c) of the Figure 3.4 marks off the part of the tracing, when there is no edge crossed. The value edge influence factor $f_r = 0$ and the withdrawal factor f_w is not changing. In the segment B, there is an important edge crossed as the edge influence value $f_r > 0$ and the value of f_w has significantly dropped. The resulting mask value f_m computed as $f_m = f_d * f_w$ is marked to the underlying mask. Processing all the fixations, we obtain the *importance mask* that serves as a vehicle for the original prioritization as recorded by the human interaction. Now, using the mask, the abstracted images a_i are assembled together forming an merged image G.

Merging Images

As mentioned earlier in 3.1, each of the abstracted a_i image has been assigned level of detail d_i. The process of creation of the merged image computes the color for every pixel in the merged image based on the local detail m obtained from the detail level mask and pixels p_i in appropriate abstracted images. For each pixel the value j is determined to correspond to the index of the abstracted image, whose detail d_i is lower than the desired detail level m. Afterwards, the f is computed denoting a relative factor of the pixel among the enclosing abstracted images a_j and a_{j+1} (We assume the $d_0 = 0$ and $d_n = 1$, where $d_i < d_{i+1}$ and n is number of abstracted images). We can write the following equations

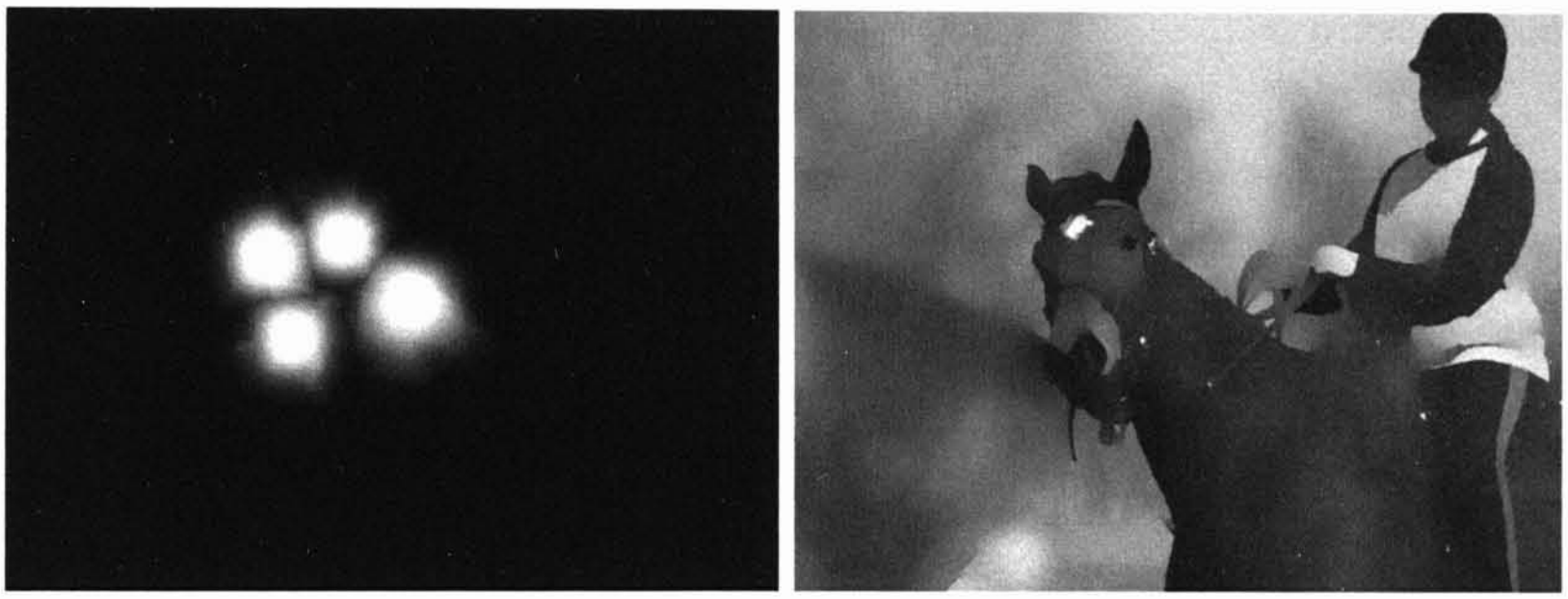
$$f = \frac{d_j - d}{d_{j+1} - d_j} \tag{3.8}$$

$$p = f * p_j + (1 - f) * p_{j+1} \tag{3.9}$$

The multiplication of the pixel p_i by a factor f corresponds to the multiplication of the values in all channels. The produced image then displays, in places with high mask value, the abstracted image with maximum detail. In the other place the less detailed abstracted image is shown. Thus, the *merged image* has non-uniform level of detail where the emphasized parts correspond to the areas identified as important by the process of human interaction.

3.3.2 Edges Tracing

As a step of the processing comes the creation of the overlayed edges. Firstly, we perform the edge detection [10] on a synthesized image. Pixels on the edges are



(a) (b)

Figure 3.6: The picture (a) shows the generated detail level mask. The white color corresponds to the maximum detail level. Processing the detail level mask with the abstracted images, the merged image (b) is generated.

then chained and the straight lines are inferred. Those are processed and merged together to form a set of results. Those might be Bezier curves [9] or straight lines. The results are then rendered to an image with transparency [10] to later draw them over the synthesized image and to accomplish the final result.

The edge detection in our work is done using the Canny edge detector [8].

This is a multistage algorithm with edge theory in background. This theory defines criteria of the optimal edge detector as:

1. *good detection* - method should find as many edges in the image as possible
2. *good localization* - found edge should be as close as possible to the edge in the real image
3. *minimal response* - every edge should be marked only once and the method should be noise resistant

The detector of Canny can be closely approximated as first derivative of gaussian[10].

As a result of edge detection, we get a binary image with the same resolution as the original one, where every pixel that belongs to an edge is marked out. The used implementation of Canny edge detector has the *low* and *high threshold* values that control the edge qualities. Lowering the values results in detection of more edges.

After the edges are detected, the pixels belonging to an edge are grouped together using the process of edge pixel chaining.

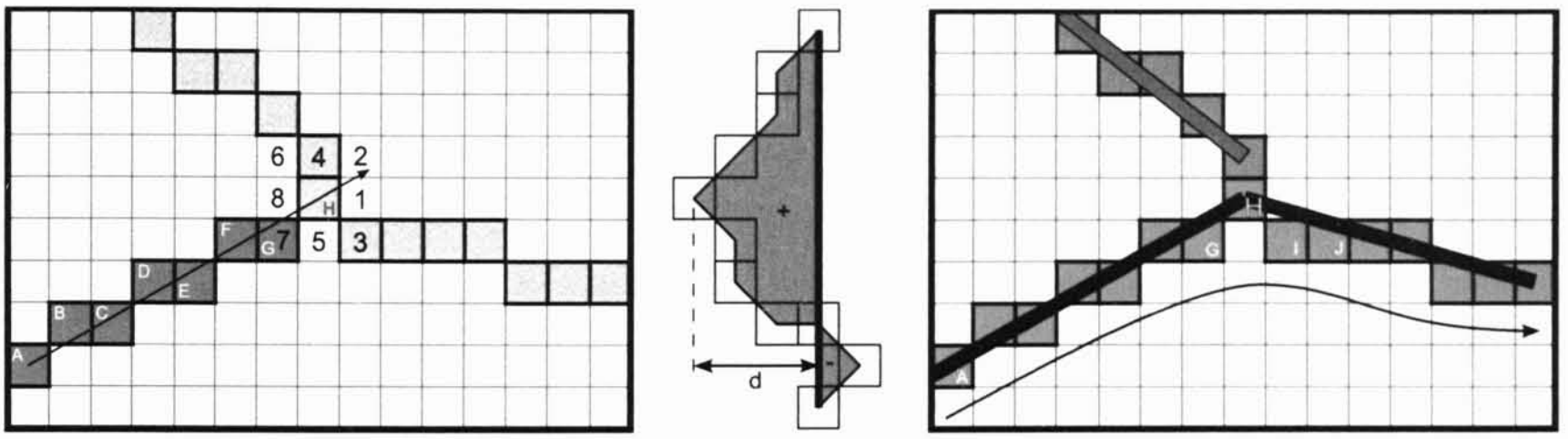


Figure 3.7: Schema of the edge chains creation process and its results.

The left picture of Figure 3.7 shows the edge pixels (colored). The edge chain then becomes a list of pixels selected in such a way that every two pixels that come one after another in a chain belong to the other eight-neighborhood [9]. In the left picture, there is a chain defined (the red colored pixels) starting with pixel marked with A and ending with pixel marked H. The orange pixel H is the last pixel of the chain, while the green pixels are not yet assigned to any chain. The process of finding the edge chains obeys the following rules:

1. Locate any pixel in the edge image that has not been visited yet. The information about a visited state is kept, disallowing the tracing algorithm to process any pixel more than once.
   In this case, the pixel marked A was chosen as first.

2. Find the neighbor pixel that lies on the same edge by searching through the eight-neighborhood.
   The pixel B was found and added to the end of the chain and became the currently processed pixel.

3. Predict the location of the next pixel by extrapolating the contained edge. If the determined pixel is a valid edge pixel, add it to the current chain, mark it as visited and repeat step 3. When the predication fails, search the neighborhood for any other pixel.
   Pixel H illustrates the situation when the edge extrapolation fails. The overlayed arrow shows the predicted direction of the traced edge. The numbers represents the order in which the neighborhood pixels are checked. The ex-

trapolation tries to keep the chains straight. In our case, it is the pixel with number 3 that is selected as the next and the tracing process continues.

4. When there are no more valid neighbors, terminate the pixel chain and try to continue with step 1. When no more pixels can be located, the chain tracing algorithm ends.

As mentioned earlier, the output of the tracing algorithm is a set of edge chains. Now, in the second phase, each chain is processed and the straight lines are extracted. One phase of the extraction process is illustrated in the middle picture of the Figure 3.7. The process itself is performed as follows:

1. The first three points from the chain are taken and the line originating at the first point and ending at the last one is formed.

2. The next point in chain is added to the line. This shifts the line end to the currently added point. The newly formed line is then evaluated. The last added point might have broken the line. The evaluation judges the quality of the line by two factors. Firstly, it is the maximum distance of the furthest point belonging to the line from the line itself. In the picture, it is the distance d. Secondly, it is the difference of the areas formed by the pixels above and under the line. This factor eliminates topologically unbalanced tracing under or above the edge. The picture illustrates those as green and red areas with the '+' (added to the area) and '−' (subtracted from the area) mark.

3. When the newly formed line does not meet the required criteria, the tracing is taken one step back to when the last point was added. The line is then stored as one of valid results of current chain processing and the tracing restarts from the end point of the just committed line.
   In the right picture, it is the point I that would have caused the line to be invalid. Thus the line $A \rightarrow B$ was stored and the tracing was restarted from the point H. The chain processing resulted in a generation of two lines, that are drawn over the pixels. (The blue line resulted from the processing of the blue pixel chain.)

Now we have extracted the straight segments of the edges and have them represented with lines. We have decided to merge the lines forming the Bezier curves to make the final state more appealing. Processing of the curves is divided into two steps.

The first step involves merging of the close pixels into the clusters (*clustering*). This allows lines—whose end points are sufficiently close to each other—to be later merged and form one compound edge. The left picture of the Figure 3.8 illustrates the clustering and the merge process. In the middle picture, all the possible results are shown. In addition to the simple line and Bezier curve, the *Bezier chains* might be created by assembling from the interconnected Bezier curves.

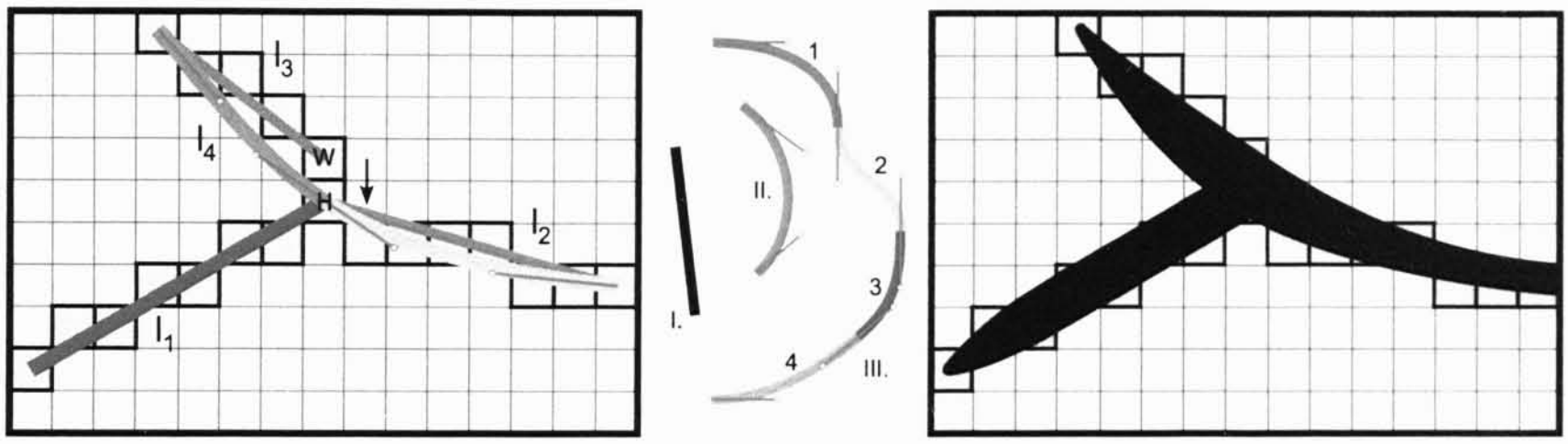


Figure 3.8: Processing and rendering of the lines traced from the pixel chains.

The grey line l_3 was previously ended in the pixel W, but due to clustering, its end was shifted to the pixel H.

The second step in the process is merging of adjacent lines to form the Bezier curves. This is done in such a way that:

1. Any line is chosen and the clusters in the beginning and in the end are inspected.

2. When any other line is connected to the appropriate cluster, the best is selected (based on its length and the containing angle) and the merge is performed. This step is repeated until any other line might be converted to the Bezier curve and appended to the beginning or to the end.

The process of clustering and merging resulted in the Figure 3.8 into two results - line l_1 and Bezier chain $\{l_2, l_4\}$.

Now when we have a collection of the trace results (lines, Bezier curves and Bezier chains). Processing continues with filtering those results to remove unsatisfactory ones. The result is rejected when it is too short or it is an boundary edge (this prevents the rendering of the frames around the image as the edges of the picture are often identified as edges (in a image processing sense).

After the processing, the results are rendered into an image which can be later drawn over the merged image (see 3.1). The rendering process draws each result individually and only controls the width of the drawn line (Bezier curve). The width is dependent on the distance to and length of the closest fixation. Given the Equation 2.3 and the closest fixation f_i with position pos_i and length len_i the width of the line in point (x, y) equals to the $angle\_to\_pixel(H(e))$. The $angle\_to\_pixel$ is a function that converts the eccentricity angle to a distance in pixels on a computer screen based on the viewer distance d and monitor resolution (Dot Per Inch). The angle e is here the eccentricity angle between the examined point and the center of the fixation. The Figure 3.9 illustrates the situation.

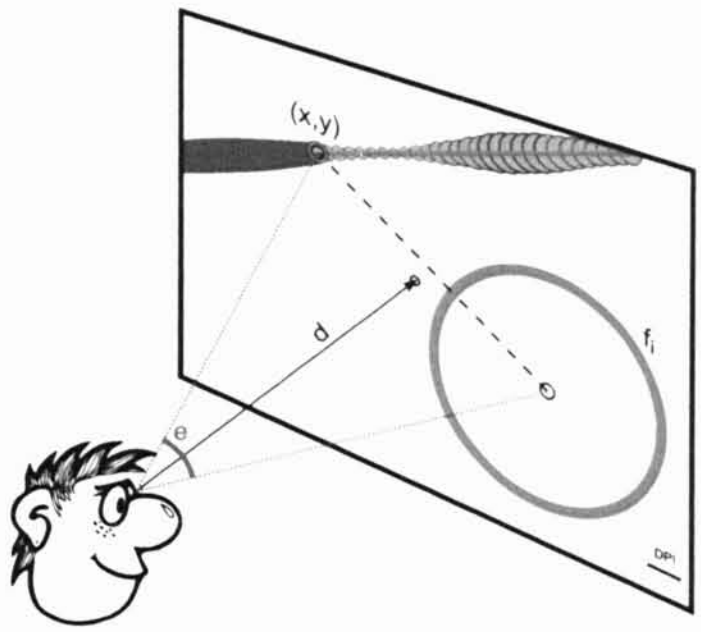


Figure 3.9: Schema of the edge point width estimation

When the width of the line is computed, the filled circle is drawn at the location of the pixel. The transparency of the circle boundaries is precisely computed resulting in fine detail. In the picture, the orange circles under the green line illustrate sizes of the individual circles as they are drawn. The result is image of rendered results (being transparent elsewhere), which might be later drawn over the merged images. The black lines drawn in the right picture of Figure 3.8 illustrate the resulting image.

Chapter 4

Implementation

The implementation of the proposed method as described in Chapter 3 consists of two separate parts. First, it is the generation of the abstracted images via the method of denoising that is implemented in MATLAB 6.5[1]. Secondly, the abstracted images are processed along with the data obtained by the human interaction resulting in the output image. This is realized as a concrete *processing case* of a *processing framework* written in Java[2].

4.1 Denoising

The implementation of the denoising in MATLAB is stored on the included disk in the directory `/matlab/`. The function $P1$, taking the name of an input image as a parameter (the name denotes the file relatively to `pictures/input/` and generating set of abstracted images with suggested parameters (Table 3.1). After loading the input image, the function $P1$ evaluates for every detail the function *process*, passing in among the input image the processing parameters that correspond to the λ, μ and number of iterations. The value pair returned contains the denoised image and the corresponding edges.

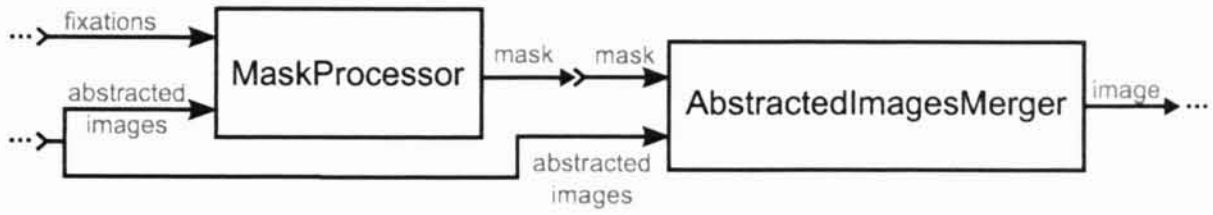


Figure 4.1: Schema of the framework processors wiring.

4.2 Processing Application

The proposed method has been implemented in the *processing framework*. The framework has been created to support the image generation process while being highly expandable. The elementary unit of the framework is the processor. As the implementation of the framework is implemented in Java, every processor is allowed to contain a code performing the target operation. The image data passed between the processors implement the `java.awt.image.RenderedImage` interface, which as being part of the Java API, might be processed by various image processing frameworks (in our case JAI[3] was used). Every processor might have defined several inputs and outputs (connectors) of a given type. The concrete processing case then specifies how the processor instances are interconnected to deliver the required results. The Figure 4.1 show the portion of the schema corresponding to the proposed method implementation. The *Mask Processor* that generates the *detail level mask* (see 3.3.1) has two input defined (marked out as arrows with name printed out in green) and one output—the generated mask image. This is connected to one of the inputs of the *Abstracted Images Merger*. The second input is connected to abstracted images, sharing the same source as the *Mask Processor*. This processor generates the *merged image* that corresponds to G in Figure 3.1.

There is another processing case supplied with the framework. Using the track and fixation drawing processors together with the image overlay processor, it generates the visual representation of the recorded track and fixations.

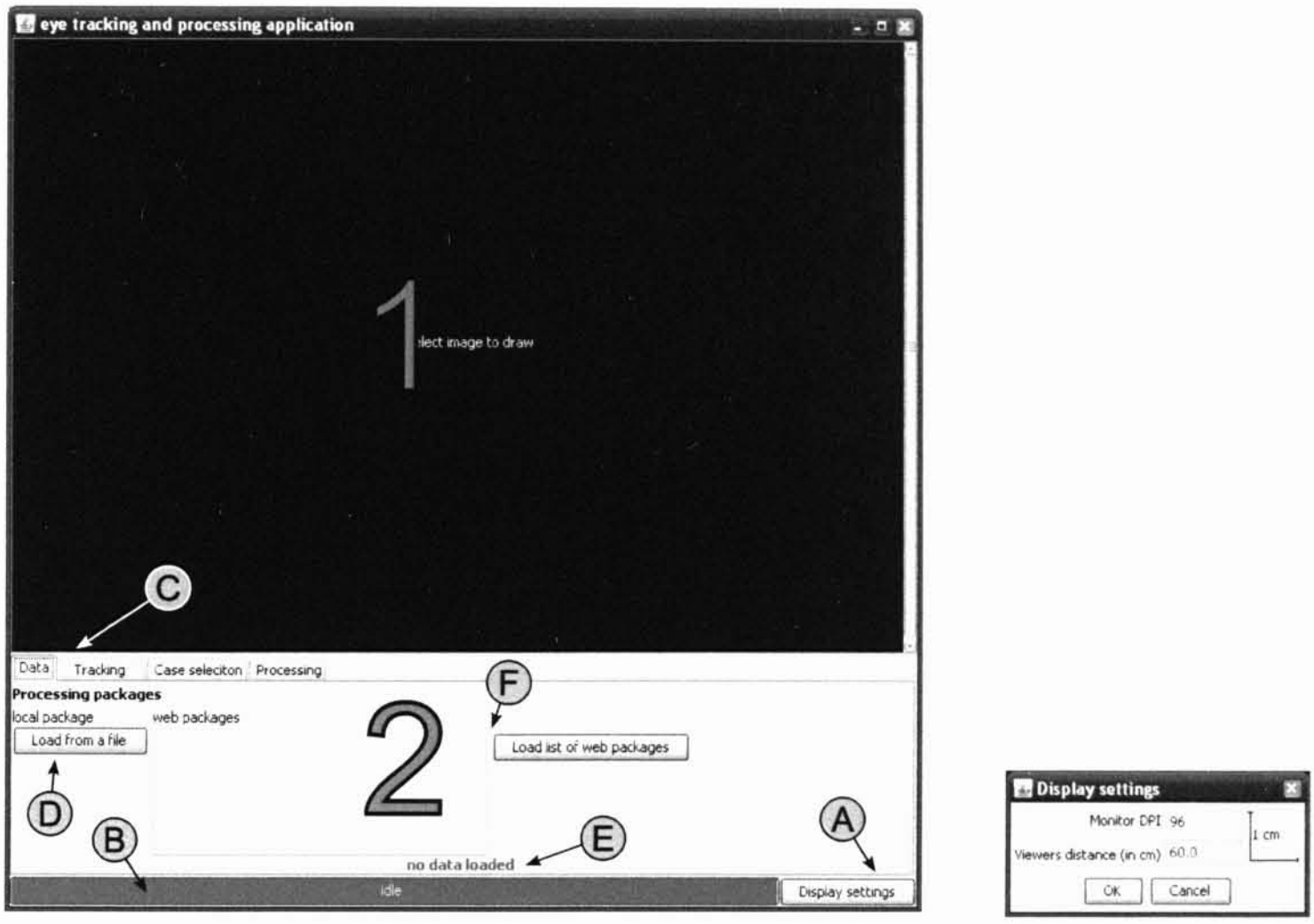
The implemented framework has a possibility to be controlled and to show the processors states. We have built the application that uses the framework and is presenting the data through the graphical user interface. We will shortly describe the application UI in action while following the typical processing scenario.

The picture (a) of Figure 4.2 shows the main window of the framework application. The area **1** displays the image and is also used in the mouse motion recording

[1]see `http://www.mathworks.com/`

[2]see `http://www.java.com/`

[3]Java Advanced Imaging - see `http://java.sun.com/javase/technologies/desktop/media/jai/`



(a) (b)

Figure 4.2: Picture (a) shows the main screen of the processing application. The window (b) allows setting-up the environment constants.

to show the examined image. In bottom part of the window **2**, all the controls are located. The button **A** shows the configuration window (b), allowing to set the environment. This includes the monitor resolution and the viewer's eye distance from the screen. At the bottom of the screen, there is a color bar **B** showing the state and process of the current operation. The tab **C** chooses the focus area of the controls presented. The currently selected area **Data** presents the controls of the framework inputs. The button **D** opens the file loading dialog allowing the user to select the Processing Package (PP). The URL of the currently selected PP is displayed at the line **E**. The application extends the framework by allowing to load PP from the remote repository as controlled by **F**.

When the PP is loaded, the tracking data have to be supplied. Figure 4.3 shows the second area of focus—**Tracking**. When the PP contains predefined tracks, they would appear in the **A** and user clicks on them, they will be loaded. The track might also be loaded/stored to the file by clicking appropriate button in a group **C**. As the track might also be recorded from the mouse motion, the record button in the **B** starts the recording process.

Figure 4.3: Tracking of the mouse motion or pre-recoded track import an the Eye Tracking panel.

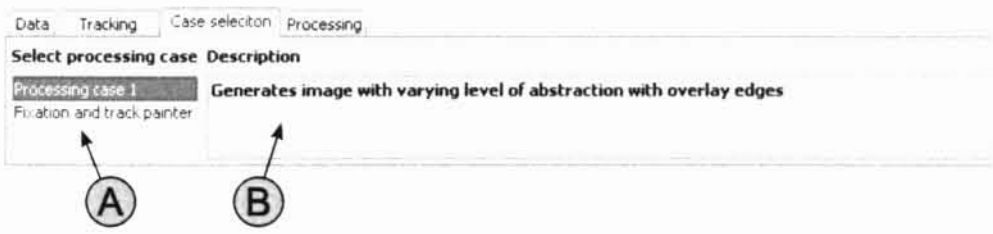
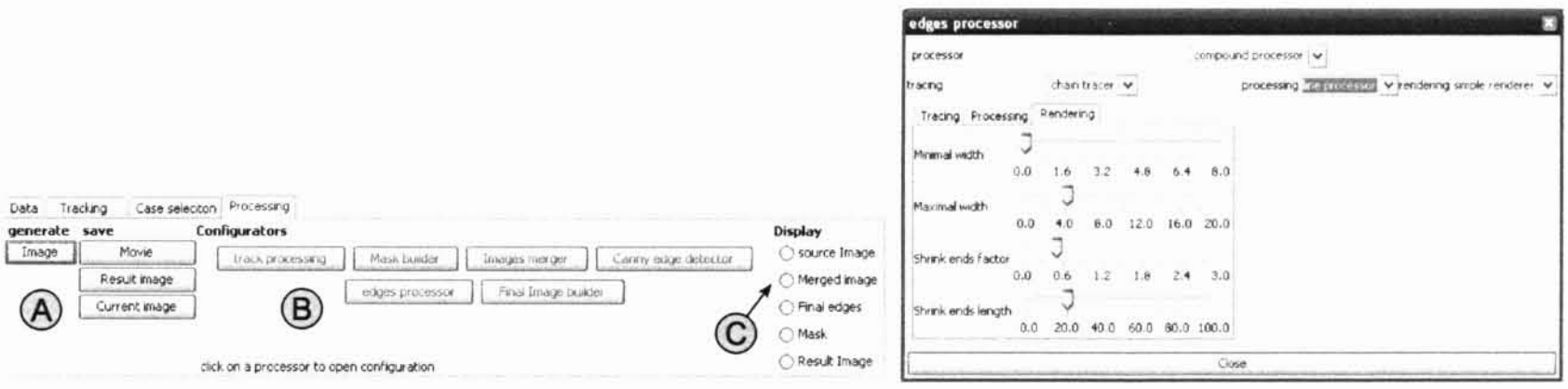


Figure 4.4: Panel with the selection of the processing case.

After all the data is ready, the processing case has to be selected. The list **A** of Figure 4.4 contains all the included processing cases and, when selected, the **B** displays the appropriate case detailed description.



(a) (b)

Figure 4.5: Overall configuration of the processing case (a) that corresponds to the proposed method. Picture (b) show the detail of the configuration of the edge tracing processor.

The last tab with name **Processing** as shown in Figure 4.5 (a) presents the available operations within the selected processing case. By clicking the appropriate button in group **A**, the resulting image might be generated/exported. Each button presented in area **B** corresponds to a particular processor. By clicking the button, the corresponding configuration window will open—as shown in (b). The selection **C** allows the user to display various inter-products of the case.

Chapter 5

Conclusion and Results

In this thesis we have tried to create a method that uses a creative process of Non-Photorealistic Rendering (NPR). It focuses on the image cognition, which is fundamental to NPR. We introduced the reader to the NPR as a field of computer graphics. The introduction is built on concrete examples from existing research. Since the method presented is not using the NPR as a goal, but rather as a vehicle for communicating, in this thesis I leave NPR behind and concentrate on its effects: *impress* or *communicate*. Having dealt with the effect, we have shifted focus to the creation process. For our purposes, we see Human Interaction as an essential part of the creation process. So we picked out specific kind of Human Interaction suitable to the method that follows. Chosen a type of Human Interaction that occurs between an image and its viewer, we identified two parts of the interaction process. First, it is the viewer responding to an image (2.3.1 Visual Perception), secondly it is the image being modified according to the viewers response (2.4 Image Abstraction). Provided the necessary foundation, we have moved on to outline the method of the presented case. Tracking the mouse motion track or importing eye-tracker data together with the Image Abstraction give us the tools to deliver our goal—effective communication through the image.

Based on the ideas from the [2] and [3] we have created the principles of a method. We allowed accessible form of human interaction—mouse to server as an input for content prioritization. The image abstraction method that we used enhances the method presented in [5] by allowing processing of the color images. The principles of the processing framework were set and the application that follows those principles was implemented. The flexible architecture of the framework

allows implementation of various method—one of the implemented ones generates the image with respect to the proposed method. The all the needed processors, that serve as reusable parts of the framework were implemented. Significant ones, relevant to the proposed method, are *Canny edge detector*, *Track processor*, *Detail Level Mask builder*, *Images Merger* and *Chain Edges Processor*. There has also been another processing case implemented, capable of drawing the recorded track and fixations (see Figure 5.1 (a)).

The proposed method successfully performs meaningful abstraction of an input image. By substituting the eye-tracker with the mouse, we made the processing highly accessible—no special hardware is required. By using the mouse, the selection of detail is more intentional—might be planned in advance. The results comply to the set goal as the identified content is emphasized at the result image. Anyway, as it is said: "*A picture is worth thousand words*". We now present the concrete example of image processed by implementation of the proposed method. Original image and more results can be found in Appendix A.

(a) (b)

Figure 5.1: Given the source image and the track (a), the image (b) was generated.

As we have found, the quality of the generated image highly depends on the quality of the abstracted images. Although, the suggested setting for abstracted images creation delivers reasonable results in most cases, we have encountered images where such parameters had to be tweaked. This especially applies to pictures with large areas of similar colors. In such case the abstracted images do not contain enough edges (are blurred to much) and the process of meaningful abstraction fails in its goal. Another area where the method might be enhanced is the edges tracking. The suggested processing sticks too much—in terms of location—with

the merged image edges. This limits the overlayed edges which when the edge middle segment faints and the edge detection fails to identify it, the traced line will split into several parts, although it is part of one edge.

Future research can enhance the proposed method in many ways. Future studies of the human visual perception can improve the abstracted images generation in a way, that the result generation will obey more the visual limitations of eye. Also incorporating the method of abstraction directly into the framework would allow controlling the abstracted images generation interactively. The concept of *detail level mask* and the flexibility of the framework allows easy exchange of the method of abstraction.

Bibliography

[1] Gooch A., Gooch B. (2001) : *Non-Photorealistic Rendering*, A. K. Peters, Ltd.

[2] Santella A., DeCarlo D. (2002) : *Abstracted Painterely Renderings Using Eye-Tracking Data.*, International Symposium on Non Photorealistic Animation and Rendering (NPAR), pp. 75–82

[3] Santella A., DeCarlo D. (2002) : *Stylization and Abstraction of Photographs.*, SIGRAPH

[4] Santella, A., DeCarlo D. (2004) : *Visual Interest and NPR: and Evaluation and Manifesto*, International Symposium on Non Photorealistic Animation and Rendering (NPAR)

[5] Šroubek F., Flusser J. (September 2003) : *Multichannel Blind Iterative Image Restoration*, IEEE Transaction on image processing, vol. 12, no. 9

[6] Cohen Y., Basri R. (2003) : *Inferring region salience from binary and gray-level images*, Pattern Recognition 36, 2349-2362

[7] Comaniciu D., Meer P. (May 2002) : *Mean Shift: A Robust Approach Towards Feature Space Analysis*, IEEE Transactions on pattern analysis and machine intelligence, vol. 24, no. 5

[8] Sonka M., Hlavac V. , Boyle R. (1999) : *Image Processing, Analysis, and Machine Vision, 2nd edition*, Brooks/Cole, Pacific Grove, CA

[9] Žára J., Beneš B., Felkel P. (1998) : *Moderní počítačová grafika*, Computer Press

[10] Gonzales, Rafael C. (2002) : *Digital Image Processing 2nd edition*, Upper Saddle River : Prentice Hall

[11] Young Paul (1987) : *The Nature of Information*, Praeger Publisher, New York

Glossary

fixation

State, when the visual gaze stays for a reasonable time on one location.

NPR (Non-Photorealistic Rendering)

area of computer graphics that focuses on enabling a wide variety of expressive styles for digital art. In contrast to traditional computer graphics, which has focused on photorealism, NPR is inspired by artistic styles such as painting, drawing, technical illustration, and animated cartoons. NPR brings together art and science, concentrating less on the process and more on the communication content of an image.

POI (Point Of Interest)

Point in a screen plane that has been captured by a eye-tracking device. The point has its *start* and *end* time the user spent looking at.

PP (Processing Package)

collection of abstracted images (with varying detail - level of abstraction), the original image and possibly prerecorded tracks packed in one file. Server as an input for a processing application.

PR (Photorealistic Rendering)

computer imaging trying to produce as realistically looking images as possible. Includes methods like 3D modeling, ray-tracing, global illumination, ...

saccade

Rapid motion of a human eye when the perception is low, because of the suppression of the cognitive processes in brain.

visual acuity

acuteness or clearness of vision, which is dependent on the sharpness of the retinal focus within the eye, the sensitivity of the nervous elements, and the interpretative faculty of the brain - Wikipedia:Visual acuity

Appendix A

Results

picture 1

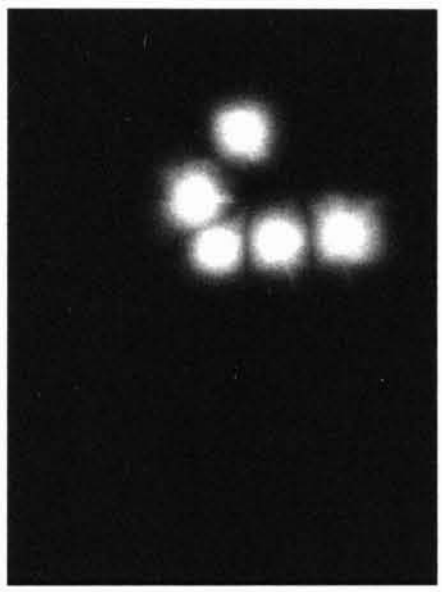
picture 2

picture 3

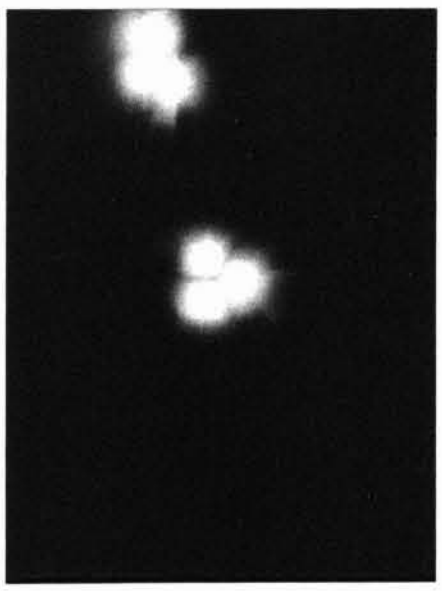
picture 4

picture 5

Processed images



processed image 2

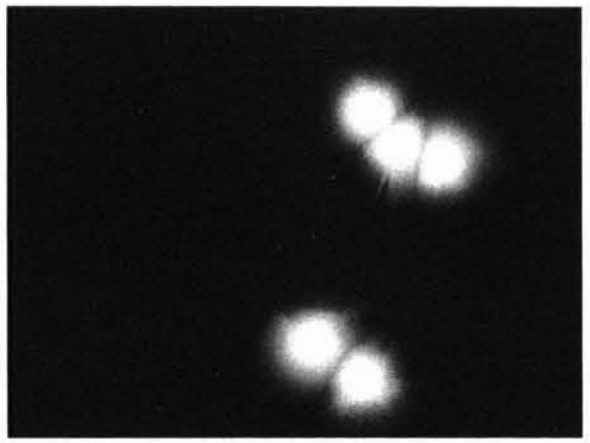


detail level map

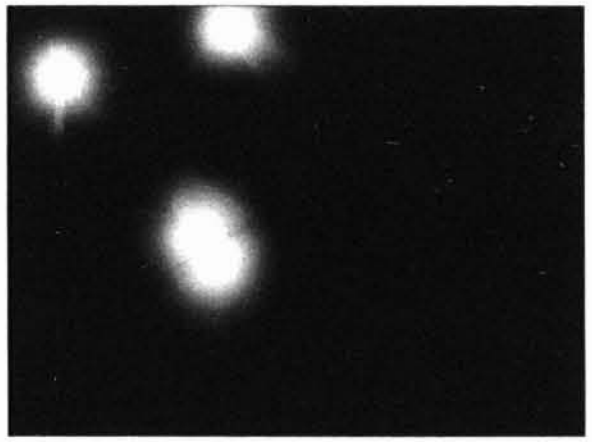
track and fixations

processed image 1

track and fixations



detail level map



processed image 3

processed image 5

Appendix B

Data format

Currently the framework supports two types of the eye-tracking data to be imported.

i. **Time bounded** - each point has it's start and end time. This delimits the period the viewer spent looking at.

t-start	t-end	x-pos	y-pos
0.0	0.016	626	178
0.016	0.055	616	158
0.055	0.079	603	141
0.079	0.086	567	144
0.086	0.102	553	148
...	...	...	...

Figure B.1: Example of the exported time bounded eye-tracker data.

ii. **Samples bounded** - each point has it's start and end sample index. The sampling period has to be known in order to determine the time the viewer spent looking at each point.

start sample	end sample	xposition	yposition
6	10	474	292
12	20	472	303
40	66	693	215
68	78	686	223
88	98	387	301
108	112	623	305
...	...	...	...

Figure B.2: Example of the exported samples bounded eye-tracker data.