Controlling Image-stylization Techniques using Eye Tracking

Maximilian Söchting and Matthias Trapp[a]
Hasso Plattner Institute, Faculty of Digital Engineering, University of Potsdam, Germany

Keywords: Eye-tracking, Image Abstraction, Image Processing, Artistic Image Stylization, Interactive Media.

Abstract: With the spread of smart phones capable of taking high-resolution photos and the development of high-speed mobile data infrastructure, digital visual media is becoming one of the most important forms of modern communication. With this development, however, also comes a devaluation of images as a media form with the focus becoming the frequency at which visual content is generated instead of the quality of the content. In this work, an interactive system using image-abstraction techniques and an eye tracking sensor is presented, which allows users to experience diverting and dynamic artworks that react to their eye movement. The underlying modular architecture enables a variety of different interaction techniques that share common design principles, making the interface as intuitive as possible. The resulting experience allows users to experience a game-like interaction in which they aim for a reward, the artwork, while being held under constraints, e.g., not blinking. The conscious eye movements that are required by some interaction techniques hint an interesting, possible future extension for this work into the field of relaxation exercises and concentration training.

1 INTRODUCTION

Motivation. The idea of an interactive system that provides users with a sensation of a *temporary* visual and aesthetic experience is not unprecedented. However, the proposed approach combines eye-tracking technology and image-abstraction techniques with a novel interaction paradigm (Figure 1). Instead of having the users be in charge of the system, this approach aims to increase the value of the visual content through instability and volatility. It aims at reversing the power dynamic that usually exists between a user and a technical device, in which the user can do almost everything with the tools he is provided with.

In the presented approach, the interactive system takes charge and prompts the user to follow a certain behavior and, if not successful, circumvent the result "reward" by nudging the user into the pre-determined behavior. Another use case of such volatile visual content is the appreciation for an evolving artwork. In contrast to still images, the interaction allows the art to become more intriguing and provide more enjoyment to observers.

Objective and Challenges. The main objective of the presented approach is to create a unique interaction experience between a user and the system, which allows the user to participate in the creation of the presented, computer-transformed visual content. It aims to create *volatile* artwork that is unique to each situation. Through the use of real-time sensors, i.e., especially eye tracking, the interaction tries to be as "frictionless" and immersive as possible while also having the option to constrain the user to a certain behavior, e.g., not blinking for a certain amount of time, as a game-like mechanic – in case it is desired by the chosen interaction technique.

As part of this work, image processing for the pur-

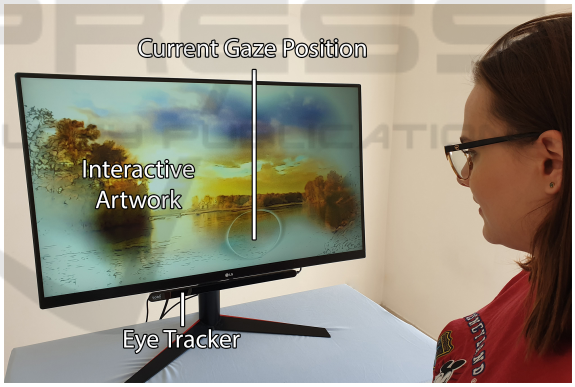

Figure 1: A user interacting with the system. The eye tracker, mounted at the bottom-side of the monitor, records the eye movement of the user. On the connected computer, the application displays the computer-generated artwork, which is influenced by the processed sensor inputs.

[a] https://orcid.org/0000-0003-3861-5759

Söchting, M. and Trapp, M.
Controlling Image-Stylization Techniques using Eye Tracking.
DOI: 10.5220/0008964500250034
In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2020) - Volume 2: HUCAPP*, pages 25-34
ISBN: 978-989-758-402-2; ISSN: 2184-4321

pose of artistic stylization is used to give the system a medium to interact with the user. Using the input from the user, the system allows for aesthetically changes in the artwork through global and local changes of abstraction parameters, technique choice, and image choice. This requires a high degree of customizability of the image-abstraction techniques to allow for the proposed interaction modes. For this work, a set of different image-abstraction techniques that imitate real mediums such as cartoon or watercolor filtering (Kyprianidis et al., 2013) have been implemented.

Contrary to the popular usage of eye trackers for statistical purposes in fields such as market research, the usage as a direct input device is not common. In a fundamental work (Zhai et al., 1999), Zhai *et al*. observed that using a solely eye tracking-based input method puts considerable strain on the user and should be combined with other input methods to allow for convenient interaction. However, since the goal of the presented work is not convenient interaction but an intensive, game-like environment with constraints on user behavior, using only eye-tracker input and the resulting user strain benefits this basic paradigm of the system by having the user "work" for his rewards.

Using real-time sensor data feeds to control the choice of image-abstraction techniques and its respective parameter values or preset poses unique challenges. Capturing, managing and provisioning multiple sensor data streams in a real-time application requires great care in the way the necessary concurrency is handled, as it is a fundamental problem of computer science. Furthermore, mapping the provided sensor data to system behavior (*interaction techniques*) in a dynamic and exchangeable manner necessitates a flexible data model and system that can provide these changes during application run-time.

Approach and Contributions. The utilized image-abstraction techniques used in this approach are based on earlier works surrounding the topic of a platform-independent format for image processing effects with parametrization on multiple levels (Dürschmid et al., 2017). Based on these approaches, a system for capturing and processing sensor data and allowing interaction techniques to be implemented as an interface between sensor data and image-abstraction was developed. Additionally, different interaction techniques have been developed and tested. To summarize, this paper makes the following contributions to the reader:

1. A concept for an interactive system that allows users to collaborate with the system in order to manipulate art work based on Graphics Processing Unit (GPU)-accelerated, aesthetic image-abstraction techniques.
2. A data format for interaction techniques and an accommodating system architecture that allows to process real time sensor data with exchangeable interaction techniques in the presented system.
3. A set of interaction techniques, their concepts, implementations and effects on users.

The remainder of this paper is structured as follows. Section 2 reviews and discusses related work with respect to interactive image processing and fundamentals of eye tracking-based interaction techniques. Section 3 presents the concept of mapping sensor data to interactions for image-abstraction, the interaction techniques, and the technical conditions required. Section 4 describes the parameterization of image and video abstraction techniques (Section 4.1) and in which way they are mapped to react to sensory inputs. Section 5 describes implementation aspects regarding the prototypical system and the image-abstraction techniques. Section 6 discusses application and results of the presented approach. Finally, Section 7 concludes this paper and presents ideas for future research.

2 RELATED WORK

Interactive Image-abstraction Techniques. As part of this work, image processing for the purpose of artistic stylization is used to give the system a medium to interact with the user (Schwarz et al., 2007). Through the input of the user, the system allows for aesthetic changes in the artwork through global and mask-based changes of effect parameters, effect choice and image choice (Isenberg, 2016). This requires a high degree of customization of the image-abstraction effects in order to allow for the proposed interaction.

Furthermore, the nature of the proposed system, including the high-frequency eye tracking device being used to manipulate the image-abstraction techniques, requires the system to be highly interactive to prevent user frustration and hold up the immersion of a fluid interaction (Vandoren et al., 2008). In order to use the described complex image stylization techniques in a real-time system, different caching and pre-loading mechanism are implemented. For this work, a set of different, complex image-abstraction techniques that imitate real mediums such as cartoon or watercolor filtering (DiVerdi et al., 2013) have been implemented in the combination with multiple interaction techniques (Section 4). Other use cases for image processing include business applications or medical analysis.

Eye Tracking Interaction. Eye tracking has a long history in medical and psychological research as a tool for recording and studying human visual behavior (Majaranta and Bulling, 2014). Eye tracking devices have been previously used in experimental and productive environments in various domains. One prominent example is the field of human-computer-interaction, in which the usage of eye trackers serves mainly two different purposes: immediate user interaction and statistical analysis.

In immediate user interaction, human-computer-interaction projects strive to utilize the gaze of the user as the only or one of multiple input components of an interactive system (Kiili et al., 2014). One such example for this mode is the work of Alonso *et al*. in the field of air traffic controllers (Alonso et al., 2013), in which the authors show that even the highly complex use of case of air traffic control can be improved upon with eye tracking technology. Still, for workloads that requires precise and quick reactions, Kasprowski *et al*. has shown that mouse or touchpad input is still superior in reaction speed and accuracy (Kasprowski et al., 2016). The more widespread purpose of eye tracking in human-computer-interaction projects is the statistical analysis. In that case, users and their gaze are observed while the user is prompted with a task, e.g., using a web site. Then, the gathered gaze data is analyzed and different conclusions can be made, e.g., which areas of the web site attract the most attention or which parts of the web site the user did not see at all. This application of eye tracking has the goal of understanding human behavior and eye movements in particular, such as in the work by Kiefer *et al*. (Kiefer et al., 2017) in which statistical eye tracking helps the researchers understand how their maps are visually processed by the test persons.

Eye Tracking in Understanding Art. In the domain of digital art creation, eye tracking has already found applications. For instance, there have been multiple art exhibitions that utilize eye tracking as an interaction technique to take control of or influence the artwork. One of these works is "Molecular Informatics" by Mikami (Mikami, 1996). In this work, users explore a Virtual Reality (VR) environment in which 3D molecule structures are automatically generated based on the gaze of the user.

Furthermore, the method of using eye tracking for statistical analysis has been used for understanding on how humans perceive and process art. One example for this is the work of Quiroga and Pedreira(Quian Quiroga and Pedreira, 2011), in which it is shown that the attention and the eye gaze of the human observers can be easily manipulated with small changes to the observed pictures.

3 TRACKING & STYLIZATION

Starting with the hardware system setup (Section 3.1), this section further describes the software system (Section 3.2), required low-level sensor data acquisition (Section 3.3), and its mapping to the different levels-of-control for image-abstraction techniques (Section 4.3).

3.1 Hardware and System Setup

In order to create a natural interaction with the system, a glasses-free consumer eye tracker (*Tobii Gaming Eye Tracker 4C*[1]) is utilized. It allows the user to influence the system as the currently chosen interaction technique intends to. The required setting for the eye tracker to work in a stable manner is described in the following (cf. Figure 1).

The eye tracker requires a well-lit environment with a single user, since the eye tracker can only track one pair of eyes at the same time. The manufacturer of the eye tracker furthermore recommends using monitors with a maximum diagonal length of 27 inches with a display ratio of 16:9 and 30 inches with a display ratio of 21:9. In addition, the distance from the sensor is advised to be kept between 50 cm to 95 cm in order to deliver consistent results. However, in the presented approach, a 43 inches 16:9 display (*Samsung The Frame*) was used with a sensor distance of approx. 1 m and found precise and stable enough for the presented system. Extension cords for the Universal Serial Bus (USB) connection of the eye tracker are advised against while the native cable has a length of 80 cm, therefore constraining the arrangement of the components in the build of the proposed system.

The eye tracker furthermore requires a calibration that tracks the head position, rotation and distance in order to produce more precise gaze tracking results. This calibration should be done on a per-user basis for long-term use, while a more casual use with multiple, quickly switching users requires a calibration-less setup for a frictionless interaction. In the latter case, a standard calibration for a certain average user position should be used. After this calibration, a guarantee for the position for the user could be achieved by suggesting such a position through ground markers or other physical constraints. Furthermore, visual feedback on whether eye detection was successful could guide the user to the right position.

[1] https://gaming.tobii.com/product/tobii-eye-tracker-4c/

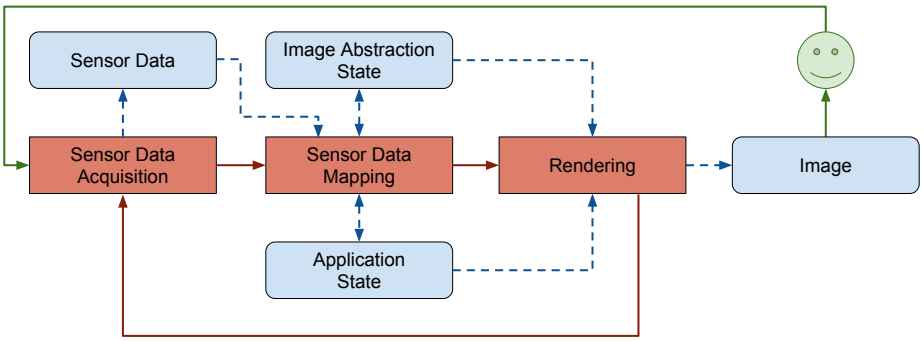

Figure 2: Overview of the software system's processing stages and control flow (red) as well as data flow (blue).

3.2 Software System Components

Figure 2 shows a conceptual overview of the software system used for our approach. Basically, it comprises three major processing stages:

Sensor Data Acquisition: This stage acquires, filters, and manages the respective sensor data from the eye tracking device and possibly other real-time sensors (Section 3.3). Its main functionality is the conversion of low-level hardware events to high-level software events required by the subsequent stages.

Sensor Data Mapping: Using the derived high-level events, this stage manipulates the respective image-abstraction state and global application state according to the different interaction mappings and modes.

Rendering: In this stage, the behavior of the active interaction modes is executed and the respective rendering and composition steps are executed to generate the complete image that is then displayed to the user.

3.3 Sensor Data Processing

The used Tobii Gaming 4C Eye Tracker provides different Software Development Kits (SDKs), such as Stream Engine, a low-level C++ binding, Core SDK, a high-level C# binding, and a Universal Windows Platform (UWP) preview of the Core SDK. Since this work is integrated into a set of related projects written in C++ and the low-level binding offers a high degree of freedom, the Stream Engine SDK was deemed most appropriate. After connecting with a linked eye tracker device, it is possible to register different callbacks, each corresponding to one data point. Every 11 ms, a sensor capture of almost every data point is generated and every corresponding callback handler registered by the developer is called. The frequency of 11 ms can not be changed programmatically, therefore the only option to reduce this, is to skip a certain amount of callbacks or synthesize callbacks together, resulting in an effective tick rate of 22 ms, 33 ms, etc.

Also, it appears that even the low-level Stream Engine Application Programming Interface (API) applies interpolation to the passed values. This can be observed when the user blinks: around 5 % of blinks done by the user do not trigger the "invalid" flag on the gaze point data point, suggesting that there may be software interpolation.

4 INTERACTION TECHNIQUES

This section describes how the parameterization of image and video abstraction techniques (Section 4.1) can be mapped, controlled, and influenced by the sensor inputs. It describes details w.r.t. the context of the proposed interaction techniques (Section 4.2) and the techniques itself (Section 4.3).

4.1 Mapping Level-of-Controls

Prior to mapping high-level sensor data events to state changes of abstraction techniques and the application itself, we briefly review the different Level-of-Controls (LOCs) offered by modern, real-time implementation of image and video abstraction techniques (Semmo et al., 2016). In particular, these are:

Pipeline Manipulation: Since an image-abstraction pipeline consists of one or more image-abstraction effects that each have global and local parameters and presets for controlling these, it is possible to define pipeline presets and pipeline parameters. However, in this work, only single effects are used, therefore eliminating the necessity of pipeline manipulation.

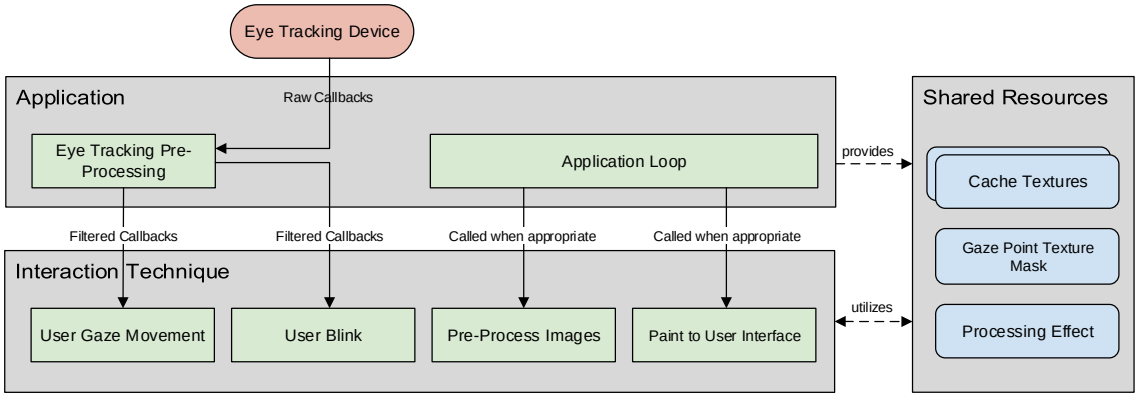

Figure 3: Overview diagram of the interaction between the application and the interaction technique (Section 4.2).

Effect Selection: Different effects are available that apply different aesthetic changes to the processed image. The selection of which effect is used can be either static (always the same effect), sequential selection (e.g., on every blink or periodically) or random selection.

Preset Selection: Each effect offers different presets for their individual set of parameters. These correspond to specific parameter value configurations that resemble a certain style and are distinctly unique between each other. The selection of which preset is used is analogous to the effect selection: fixed selection, sequential selection, random selection or no selection at all. In the last case, the default preset will be automatically selected for each effect.

Adjusting Global and Local Parameters: Effects possess parameters which are used during processing to adjust the behavior of the effect. They can be numerical, floating point, natural numbers, or an enumeration values. These parameters can be changed globally, i.e., for every pixel in the image, or locally, i.e., for only a subset of the pixels within the image, usually adjusted with mask painting (Semmo et al., 2016).

4.2 Interaction Context

Given an input image and an image-abstraction technique which parameter values can be locally controlled using masks (value encoded in grey-scale, and direction as 2D vector). The abstraction technique generates the artwork. An eye tracker mounted on the display is used to capture the (1) the eye position p, (2) the gaze movement $d(p)$ and (3) blinks over time.

Based on these values, the active abstraction technique can achieve its desired behavior (Section 4.3) through executing custom behavior in four different points of time (Figure 3). Given their nature, interaction techniques are implemented as a Strategy pattern (Gamma et al., 1995).

Gaze Movement: As an abstraction to the raw callbacks provided by the eye tracker API (Section 3.3), the abstraction technique is provided with an interpolated, to the application window adjusted gaze position every time a batch of these is processed (2 to 4 times of the sensor interval, i.e., 22 ms to 44 ms). This position is by default used to paint a circle shape onto the Gaze Movement Texture at the detected gaze point. However, this default behavior can be extended with custom processing, which utilizes the gaze position.

Blink: The Blink event is provided by the application whenever the user blinks. This is determined through a confidence model that analyzes the raw gaze movement callbacks and their respective validity (Section 5.3). The default behavior, which can be extended analogously to the other events, causes the Gaze Movement Texture to be reset. This is a behavior common to the design of many of the implemented interaction techniques, which in turn provides a common fundamental understanding of the behavior of the system to the user ("blink = advance, loss", Section 6).

Image Preprocessing: This stage prepares for the painting to the user interface by rendering the required abstract image(s) into the two provided Cache Textures. Pre-processing these images eliminates the need to render these multiple times or even in real-time, which would impede the application performance significantly. In order to execute the desired behavior (Section 4.3), the interaction technique accesses the Processing Effect and applies changes to the effect configuration on

(a) Spotlight mode. (b) Shift mode. (c) Coloring mode.

Figure 4: Overview of different interaction modes. The current tracked gaze location is depicted by a circular shape.

different levels (Section 4.1), e.g., applies parameter presets or changes the active effect or image, followed by rendering the effect image(s) to the Cache Texture(s).

Rendering to User Interface: The last stage, Paint to User Interface, is called once the main window is prompted for redrawing, i.e., the user interface and the user-side representation of the artwork have to be redrawn. Typically, the two Caching Textures and/or the Gaze Movement Texture are being drawn in a specific order with specific composition modes to achieve the desired effect. However, this stage can also be as simple as just drawing one of the Cache Textures.

In order to keep state information between the different events and follow the designated program logic, the following global resource slots are provided to each callback:

Cache Textures: In order to pre-process and cache the abstracted images, two textures are provided by the program. They are automatically resized to the current rendering resolution and stored on the GPU in order to reduce transfer between GPU and Central Processing Unit (CPU) during the Paint to User Interface phase. Having two instead of one texture allows for interesting abstraction techniques that utilize blending between presets within the same effect or between different effects. Extension to more than two textures is trivial, but was not necessary for the implemented abstraction techniques.

Gaze-movement Texture: The Gaze Movement Texture is by default managed by the application (see previous paragraph) and contains a gray-scale mask of the previous gaze movement of the user since the last time he blinked. It is automatically reset once the user blinks and utilized during the image preprocessing phase to generate compositions with the abstracted images.

Processing Effect: The Processing Effect is the interface to the rendering core. It allows to change the currently active effect, the currently active preset and single parameters. In theory, it also also allows the combination of effects in a pipeline, however, this functionality was not used within the scope of this work.

4.3 Operations Modes

In the following, a *selection* of interaction modes are presented that follow similar design principles but represent unique experiences (cf. Figure 4).

Spotlight-mode. Starting with a solid color (black) or with an extremely blurred input photo, the artwork is revealed successively using gaze movement (Figure 4a). Therefore, an alpha mask, which blends the abstracted image with the canvas, is manipulated by in-painting a circle or similar shape at the position where the user's eye is centered. If the user blinks (e.g., a certain number in a certain period of time), the mask is cleared and the revelation process must be performed from the beginning.

Shift-mode. Using the functionality of the Spotlight-Mode, the alpha-mask blends between different level-of-abstractions, filtering presets (Figure 4b). Given different level-of-abstractions or different parameter combinations represented by presets (predefined parameter values), the mask follows the eye movement yielding – for example – low abstraction (or the original image) at the gaze focus area and high abstraction in the remaining area of the image. This implies two aspects: (1) the artwork becomes dynamic and unique and (2) a user never sees the complete image. As an extension, the system can record the mask values and generate a video for sharing.

Coloring-mode. Starting with a light outline or pencil sketch of the original image, gaze movement

will blend to a colored version of the same image at the gaze location (Figure 4c). Implementation-wise, the Coloring-Mode is a special case of the Shift-Mode, in which two presets of an image-processing technique that imitates watercolor painting are chosen. One of these presets produces color-less pencil sketches while the other produces softly colored watercolor paintings of the original image. Blending these two presets results in the desired effect of "coloring-in" the image.

5 IMPLEMENTATION ASPECTS

In this chapter, the implementation of the preceding concept is discussed. First, on overview of the system architecture is given (Section 5.1) and the implementation of the image-abstraction techniques is presented (Section 5.2). Finally, the analysis and processing of the eye tracking data is explained (Section 5.3).

5.1 Overview of System Architecture

In Figure 5, an overview of the system architecture is shown. The *MainWindow* utilizes an User Interface (UI) Toolkit and relying on respective widgets and rendering management. *CanvasWidget* implements most of the application logic as it described in the previous chapters. It manages the creation and handling of the Interaction Technique by providing it with the respective events and resources (Figure 3). As part of that, it collaborates with the Effect Processor and handles the Processing Effect object that is mainly accessed by the Interaction Technique. The EyeTracking module takes care of the batching, analysis and processing of the eye tracking sensor data (Section 5.3) and communicates with the connected Tobii eye tracker through the Tobii Stream Engine library.

5.2 Image-abstraction Techniques

The image-abstraction techniques are implemented using a platform independent effect format, which allows parametrization on local and global levels that has been developed in previous works by Dürschmid *et al*. (Dürschmid et al., 2017). Another goal of the format is the separation of the platform-dependent and platform-independent parts, separated into four different categories.

Effect Definition: The effect definition describes the parameters and the parameter values of the presets of the effect. It also links to an implementation set that implements this effect.

Implementation Set: The implementation set describes how an effect is executed on different platforms by linking multiple implementations that each correspond to an operating system or graphics API requirement.

Implementation: The implementation describes how the effect is executed in a specific environment, e.g., following an operating system or graphics API constraint. It describes the shader programs and references the specific shader files, which should be executed.

Common Assets: These include preset icons, canvas textures and shader files.

For the execution of these effects, a C++ processor that supports this format is utilized as a library for this project.

5.3 Analysis of Eye Tracking Data

We distinguish between processing of high-level and low-level events as follows.

Low-level Eye-tracking Events. The received, low-level eye tracking data is analyzed before being used to control the image-abstraction technique. First, the raw callbacks of the Stream Engine API are collected and saved in batches. Then, these batches are processed collectively after a certain time frame. For every valid gaze point, a circle shape is, by default, painted onto the provided Gaze Movement Texture at the detected gaze point (Section 4.2). The gaze point can also be used for custom processing, if the interaction technique has implemented such behavior. The mask is provided to each interaction technique, which utilize it in different ways, e.g., as a clipping mask, however all of them reset the texture mask once the user blinks.

High-level Eye-tracking Events. In order to derive whether the user has blinked or not, a basic confidence model is used. This model analyzes the amount of gaze point callbacks that deemed their respective measurement as invalid, therefore hinting the absence of a valid pair of eyes. This improves the stability of the blink detection significantly, since even during stable measurement periods, occasional invalid callbacks may occur, which would in turn produce erroneous blink events.

Fixations, i.e., sustained gazes of the same point, are derived during gaze point processing. For this, the

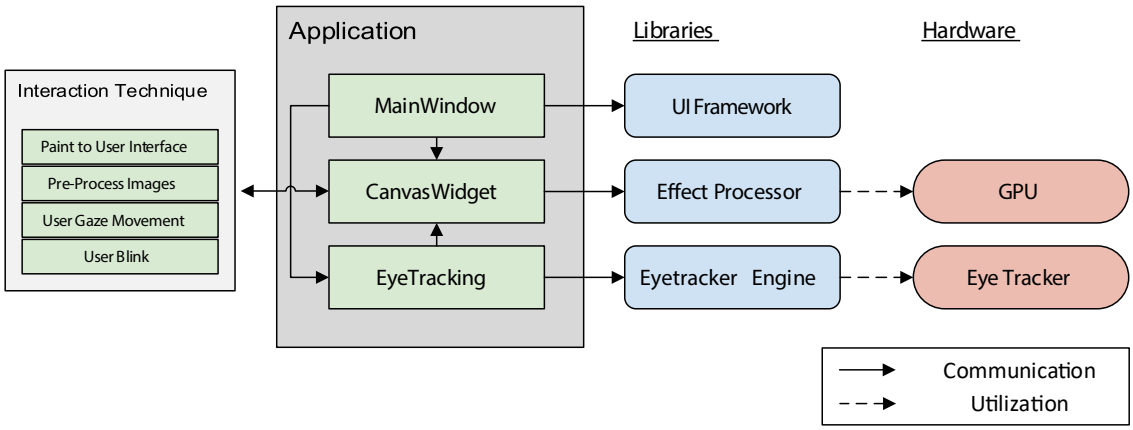

Figure 5: Overview of the system architecture and external dependencies. The three components of the application collaborate with external libraries and, transitively, with the respective hardware, to implement the presented concept.

Manhattan distance between the last n gaze points is computed. If it does not exceed a certain threshold, a fixation is assumed. Once the current gaze point moves a sufficient amount and exceeds the distance threshold, the fixation is finished and regular gaze point processing commences.

6 RESULTS AND DISCUSSION

This section describes an examplary prototypical application of the proposed system, discusses conceptual and technical challenges, and present future research ideas.

6.1 Application Scenario

With the advent of the digital age and the emergence of smart phones equipped with photo sensors, acquisition, provisioning, and consumption of digital visual media such as images or videos has become incredibly frictionless. Together with the development of low-cost, highly accessible mobile data infrastructures in the majority of western countries, sharing images and videos has become one of the most dominant components of global bandwidth usage. In this mobile age, social networking apps such as Instagram or Snapchat are centered around communication using images and short videos – such digital visual content has become a major part of modern digital life.

However, with this development, the consumption of visual media has become arbitrary. The acquisition and sharing of such media has turned so common and casual that it has become an almost mandatory task to express oneself in this social age frequently while older content is rarely looked at. This yields the hypothesis that the frequency in which visual media content is created seems more important than the represented content itself.

From this loss of appreciation for the media, the idea of an interactive system that provides users with a sensation of a *temporary* visual and aesthetic experience arose. Instead of having the users be in charge of the system, this approach aims to increase the value of the visual content through instability and volatility. It aims at reversing the power dynamic that usually exists between a user and a technical device, in which the user can do almost everything with the tools he is provided with. However, in this approach, the interactive system takes charge and prompts the user to follow a certain behavior and, if not successful, takes away from the "reward" for the user, nudging the user into the pre-determined behavior. Another use case of the volatile visual content is the appreciation for an evolving artwork. In contrast to still images, the interaction allows the art to become more intriguing and provide more enjoyment to observers.

The main objective of the presented approach is to create a unique interaction experience between a user and the system that allows the user to participate in the creation of the presented, computer-transformed visual content. It aims to create a *volatile* artwork, which is unique to each situation. Through the use of sensors, such as eye tracking, the interaction tries to be as "frictionless" and immersive as possible while also having the option to constrain the user to a certain behavior, e.g., not blinking for a set amount of time, as a game-like mechanic – in case it is desired by the chosen interaction technique.

6.2 Discussion

With the proposed system architecture, it was possible to implement the presented concept. Together with the introduced system components, i.e., the Tobii 4C Eye Tracker and the Samsung "The Frame" TV, it is possible to execute the presented interaction techniques with an interactive frame/response rate. During application run-time, minor lag ($<$2 s) occasionally occurs during blink events since in that point of time, images may be pre-loaded from the hard drive and the effect pipeline may be rebuilt to contain a different effect. However, since the user is blinking at that time, this lag is less noticeable, especially when using resource-light interaction techniques.

An interesting observation that occurred during the development of the Revelation-Mode (Section 4.3) is the necessity of highly conscious eye movements. Usually, the gaze point of eyes is determined by visual interest (interesting objects or patterns) while much information can already gathered from peripheral vision. However, the proposed interaction technique Revelation-Mode requires users to repeatedly look at large regions of blackness, going strongly against the normal behavior of human eye movement. Even after the user has learned how the system works, the expectation of the system to react based on that does not make the eye movements feel more "natural". The repeated conscious movements may be perceived as exhausting when interacting with this technique for longer amounts of time because of the unfamiliarity. Interestingly, conscious eye movements like these are also used in relaxation exercises and even have been trialed for the therapy of mental health conditions (Vaughan et al., 1994).

For all the other interaction techniques, the experienced discomfort is significantly smaller since there is no necessity to look at seemingly "nothing". However, the notion of using eye-tracker as an input device and triggering direct effects in the presented interface still feels unfamiliar. Most likely, this is because everyday digital devices such as consumer Personal Computers (PCs) and mobile devices operate with touch or mouse/keyboard input, allowing the eyes to look at arbitrary points in the interface without triggering any side effects.

Through the reuse of certain interaction patterns in most interaction techniques, a common understanding of the interaction principles of the system is being formed. With time, the user picks up the interaction sequences and starts to associate these with their respective meaning. Such an example would be that blinking usually causes a reset in progress and a change in the style of the displayed artwork. Also the fundamental interaction of causing a certain influence onto the displayed picture by directing the gaze of their eyes is quickly learned and highly intuitive, while also infuriating in some parts, since nothing in the artwork can be fixated without it transforming into something else.

6.3 Future Research Directions

For future work, it is possible to complement the current sensor inputs with additional sensors to facilitate interaction techniques utilizing various inputs. Examples for such complementing sensors can be microphones measuring acoustic pressure or reacting to voice commands, an ambient light sensor that influences the colors in the artwork, and wearables such as smart watches that could transmit the heart rate of the user.

Furthermore, an extension of the interaction techniques to allow for more dynamic shared resources could be implemented to enable more complex interaction modes that may even use 3rd-party APIs or libraries for additional sensor or miscellaneous data. Also, productive use-cases in the medical domain, such as relaxation exercises and concentration training, could be approached by extensions of the presented approach. Specific interaction techniques that make the user follow certain patterns with their eyes could imitate existing exercises and vision therapies.

7 CONCLUSIONS

This paper presents a novel approach for interactively controlling image-abstraction techniques at different Level-of-Control (LOC) using eye tracking sensor data. By mapping of eye movement and blinking to global and local LOC, the presented system enables different interaction techniques based on similar design principles, allowing the user to learn the principles of the interaction with the system.

REFERENCES

Alonso, R., Causse, M., Vachon, F., Parise, R., Dehais, F., and Terrier, P. (2013). Evaluation of head-free eye tracking as an input device for air traffic control. *Ergonomics*, 56:246–255.

DiVerdi, S., Krishnaswamy, A., Mech, R., and Ito, D. (2013). Painting with polygons: A procedural watercolor engine. *IEEE Transactions on Visualization and Computer Graphics*, 19:723–735.

Dürschmid, T., Söchting, M., Semmo, A., Trapp, M., and Döllner, J. (2017). Prosumerfx: Mobile design of image stylization components. In *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*, SA '17, pages 1:1–1:8, New York, NY, USA. ACM.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Isenberg, T. (2016). Interactive npar: What type of tools should we create? In *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering*, Expresive '16, pages 89–96, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Kasprowski, P., Harezlak, K., and Niezabitowski, M. (2016). Eye movement tracking as a new promising modality for human computer interaction. In *2016 17th International Carpathian Control Conference (ICCC)*, pages 314–318.

Kiefer, P., Giannopoulos, I., Raubal, M., and Duchowski, A. (2017). Eye tracking for spatial research: Cognition, computation, challenges. *Spatial Cognition & Computation*, 17(1-2):1–19.

Kiili, K., Ketamo, H., and Kickmeier-Rust, M. (2014). Evaluating the usefulness of eye tracking in game-based learning. *International Journal of Serious Games*, 1(2).

Kyprianidis, J. E., Collomosse, J., Wang, T., and Isenberg, T. (2013). State of the art: A taxonomy of artistic stylization techniques for images and video. *IEEE Transactions on Visualization and Computer Graphics*, 19(5):866–885.

Majaranta, P. and Bulling, A. (2014). *Eye Tracking and Eye-Based Human–Computer Interaction*, pages 39–65. Springer London, London.

Mikami, S. (1996). Molecular informatics. morphogenic substance via eye tracking. `https://bit.ly/2oezCZv`.

Quian Quiroga, R. and Pedreira, C. (2011). How do we see art: An eye-tracker study. *Frontiers in Human Neuroscience*, 5:98.

Schwarz, M., Isenberg, T., Mason, K., and Carpendale, S. (2007). Modeling with rendering primitives: An interactive non-photorealistic canvas. In *Proceedings of the 5th International Symposium on Non-photorealistic Animation and Rendering*, NPAR '07, pages 15–22, New York, NY, USA. ACM.

Semmo, A., Dürschmid, T., Trapp, M., Klingbeil, M., Döllner, J., and Pasewaldt, S. (2016). Interactive image filtering with multiple levels-of-control on mobile devices. In *Proceedings SIGGRAPH ASIA Mobile Graphics and Interactive Applications (MGIA)*, pages 2:1–2:8, New York. ACM.

Vandoren, P., Van Laerhoven, T., Claesen, L., Taelman, J., Raymaekers, C., and Van Reeth, F. (2008). Intupaint: Bridging the gap between physical and digital painting. In *2008 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*, pages 65–72.

Vaughan, K., Armstrong, M. S., Gold, R., O'Connor, N., Jenneke, W., and Tarrier, N. (1994). A trial of eye movement desensitization compared to image habituation training and applied muscle relaxation in post-traumatic stress disorder. *Journal of Behavior Therapy and Experimental Psychiatry*, 25(4):283 – 291.

Zhai, S., Morimoto, C., and Ihde, S. (1999). Manual and gaze input cascaded (magic) pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 246–253, New York, NY, USA. ACM.