東京大学大学院新領域創成科学研究科
複雑理工専攻

修士論文要旨

眼球サッケード運動計測による映像のレンダリング効率化
(A method of rendering optimization based on saccade detection)

Key Words: Saccade, eye tracking, rendering

2017 年 3 月 10 日修了

指導教員　篠田　裕之　教授
　　　　　牧野　泰才　講師

47-156114
杜　洪林

A Method of Rendering Optimization Based on Saccade Detection

Honglin Du   Shinoda Makino Laboratory

1. Introduction

Saccade refers to a rapid jerk-like movement of the eyeball which subserves vision by redirecting the visual axis to a new location. Saccadic suppression[1], also known as saccadic masking, is the phenomenon in visual perception where the brain selectively blocks visual processing during eye motions, in this way, neither the motion of the eye nor the gap in visual perception is noticeable to the viewer.

In this research we propose a new method of rendering optimization based on saccade detection. We use eye-tracking method to detect the onset of a saccade in real time and briefly stop the rendering process, then reopen it before the saccade ends. In this way computing resources can be reduced and an improvement of computer performance can be achieved. We also design a psychophysical experiment to evaluate the proposed method and have a discussion about the experiment results.

2. Prototype System Implementation

A prototype system has been implemented in this research. It mainly consists of two parts which are software and hardware. All elements are categorized to realize three different functions: Saccade detection, reflection elimination and scene presentation.

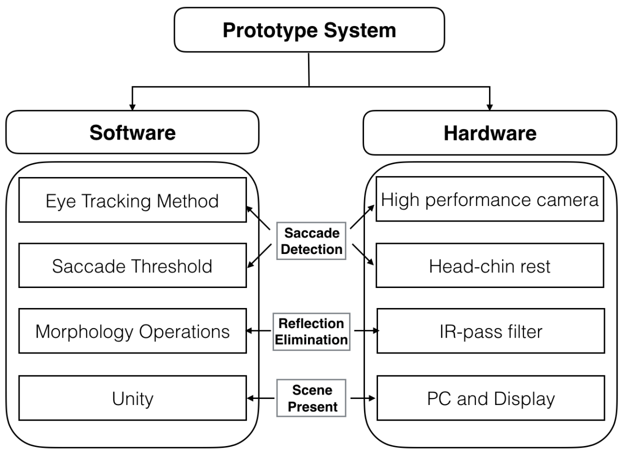

Figure 2.1

Since eye-tracking is necessary to saccade detection[2]. A feature-based approach for eye centre localization that utilizes image gradients has been used in this system. It can accurately and efficiently detect and track eye centers in low resolution images and videos[3].

As for the detection error during process, we decide to use a threshold value to help detect the onset of a saccade. If the distance between two consecutive eye localization is bigger than the threshold value, a saccade is considered to be detected. Otherwise the eye will be regarded as still. In order to find the proper threshold value, we record a video while the saccade detection process, then find the moments that detection error happens, which means detection result changes erratically while subject's focus stays still. According to the distance data of those unstable moments, a threshold value is determined.

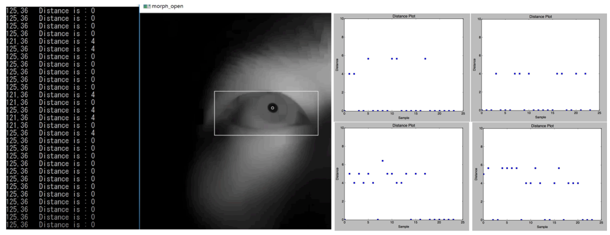

Figure 2.2

Figure 2.2 shows several different errors. Mainly there are 3 types of distance errors that are 4, 5 and 5.65685 respectively. However in very rare situations it comes to 6.40312 which happened only once in the test video. Therefore we decide to use 6.5 as our detection threshold value.

Saccade detection needs to be as fast as possible so a high performance camera is necessary. Here we use Ximea MQ003MG-CM. It has a USB3 interface, fast process capturing to make sure its FPS could reach more than 500 at most. In order to reduce the reflections caused by eye structures IR filter and image processing operations have also been used.

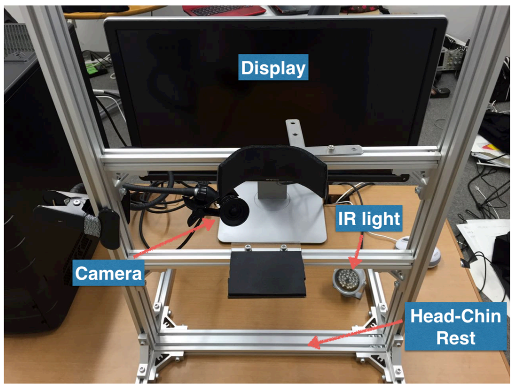

Figure 2.3

The hardware layout is shown in figure 2.3. This system comprises a Dell P2415Q(DP) 24" 3840x2160 IPS LCD monitor, a Ximea MQ003MG-CM high performance camera, a head-chin rest, an IR light, a Gtune desktop PC with Windows10 operation system, an Intel Core i7-6700(3.8Ghz) CPU, and a NVIDIA GeForce GTX 1080 GPU.

3. Experiment

After the implementation of eye localization method and with all the hardware set up, here is the experiment procedure. The purpose of this experiment is to verify saccade suppression effect during saccadic rendering and evaluate the implemented prototype system.

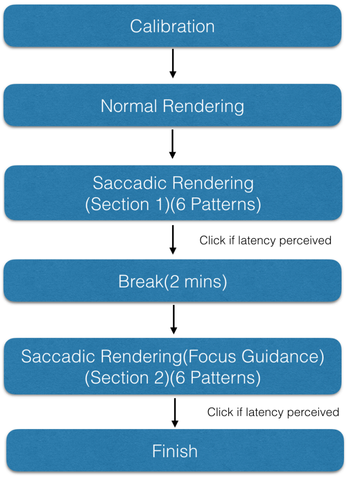

Figure 3.1

The experiment procedure mainly consists of two sections. In each section, while subject is watching the animation, eye images will be recorded by the high performance camera and eye centers are localized by system in real time. Once the distance between two consecutive eye centers is larger than the threshold value, a saccade is detected. The rendering will be instantly paused for a very short time and this time(latency), has 6 patterns that are 20ms, 40ms, 60ms, 80ms, 100ms and 120ms respectively. Each of the patterns lasts for 60 seconds. The only difference between these two sections is that animation in section 2 is presented with focus guidance. A pointer is generated during the animation and it will randomly move among 5 different spots. Subject is told to focus on the pointer only and also click the mouse if any animation latency is perceived at any time.

	Section 1			Section 2		
Latency	Saccade	Click	Ratio	Saccade	Click	Ratio
20 ms	669	23	3.44%	602	12	1.99%
40 ms	529	73	13.78%	511	36	7.05%
60 ms	469	102	21.75%	423	70	16.55%
80 ms	447	210	46.98%	381	168	44.09%
100 ms	399	247	61.90%	374	221	59.09%
120 ms	366	260	71.04%	345	230	66.67%

Table 3.1

Table 3.1 is total saccade and click amount for all 8 subjects. It shows click amount, which means the amount subject perceived latency, increases along with the growth of latency. And saccade amount in section 1 is generally larger than saccade amount in section 2.

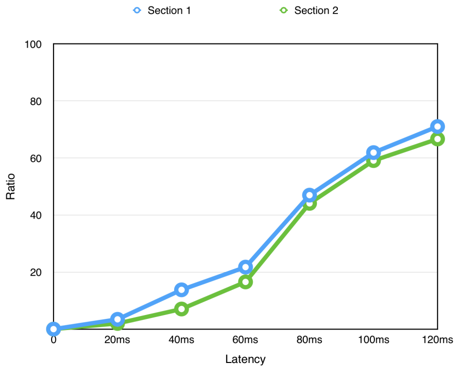

Figure 3.2

Figure 3.2 shows the comparison result of ratios in different sections. Through all types of latency, perception ratios of section 2 are lower than those in section 1. Also, for both sections when rendering latency reaches to 80ms perception ratio rises rapidly. This indicates subjects start to have a strong perception of latency. Therefore 60ms is proved to be the optimal value to the prototype system. According to this result, The total saccade amount in section 2 of 60ms-latency, for example, is 423 for 8 subjects. Take an average as 53, multiplied by 60ms the result is 3.18 seconds. As one pattern lasts 60 seconds we can easily figure out that saccadic rendering saves 5.3% of graphics computation resources.

4. Summary

In this study, a new method of rendering optimization based on saccade detection is proposed. To implement a prototype system, we used high performance camera to record eye images in high fps, utilized both hardware and software to reduce eye reflections, and found a proper threshold value to detect the onset of a saccade. We also designed an experiment to evaluate this prototype system. This experiment was conducted with 8 subjects and the experimental data was analyzed as well.

According to the results, saccade suppression effect is verified during saccadic rendering. And 60ms is proved to be the optimal rendering latency of this prototype system. By using this latency, the system can save around 5.3% of computation resources.

5. Reference

[1] Erdmann B, Dodge R. Psychologische Untersuchungen über das Lesen auf experimenteller Grundlage[M]. Books on Demand, 1898.

[2] Hansen D W, Ji Q. In the eye of the beholder: A survey of models for eyes and gaze[J]. IEEE transactions on pattern analysis and machine intelligence, 2010, 32(3): 478-500.

[3] Timm F, Barth E. Accurate Eye Centre Localisation by Means of Gradients[J]. VISAPP, 2011, 11: 125-130.