Chapter 1
Image Analysis

Detlev Droge
University of Koblenz-Landau, Germany

ABSTRACT
This chapter focuses on the image processing part of eye tracking systems. Basic knowledge of image processing is assumed. After an overview of the possible input images and some remarks on preprocessing of the images, we will focus on the detection relevant features such as pupils and glints. The last part of this chapter focuses on estimating positions of these features. It is not possible to present a comprehensive solution for an eye tracker in this chapter; however, we will indicate possible yet simplified methods in the different steps of processing and demonstrate how images can be processed to obtain real-time performance. The program code is given in Matlab (Octave) language for clarity.

INTRODUCTION
This chapter focuses on the image processing part of eye tracking systems. Basic knowledge of image processing is assumed. The reader might want to refer to books established as references in this field, such as those of Jain (1989) and Jähne (2005), for more in-depth background.

The organisation of this chapter is as follows. After an overview of the possible input images in “What is in the image?”, and some remarks on “Preprocessing”, we will focus on “Detection relevant features” such as pupils and glints. The last part of this chapter focuses on “Estimating positions” of these features.

As almost always with real-time tasks, efficiency is to be considered for all steps, since image processing is a computationally expensive task. As the user’s system usually not only runs the eye tracking software but primarily handles the user’s application programs, the tracking system must not consume all available CPU resources. An extensive overview of the topic of eye detection and the algorithms published can be found in the work of Hansen and Ji (2010). Only some of them are mentioned in this chapter.

DOI: 10.4018/978-1-4666-3994-2.ch001
It is not possible to present a comprehensive solution for an eye tracker in this chapter; however, we will indicate possible yet simplified methods in the different steps of processing and demonstrate how images can be processed to obtain real-time performance. The program code is given in Matlab (Octave) language for clarity.

WHAT IS IN THE IMAGE?

Depending on the eye tracker set-up, the camera usually provides images of either faces or only eyes. Systems working with eye images allow for more accuracy while those showing the whole face allow for a larger range of head movements. Head-mounted eye trackers have the camera very close to the eye and usually produce images showing the eye only, similarly to Figure 1 (a). Remote eye trackers are generally much less intrusive; however, this imposes some costs. Using a camera with a narrow field of view (FOV) requires the user to keep the head in a rather fixed position. The resulting images are similar to those from head-mounted devices. A wide FOV is needed to allow greater freedom of head movement. Consequently, the camera images show the whole face as in Figure 1 (b) and – most importantly for the image processing – provide only much lower resolution to the image of the eye, making it more difficult to acquire accurate measurements.

Another question related to the set-up of the eye tracking system that has considerable impact on the image processing is the position of the light source(s) with respect to the optical axis of the camera. These (usually infrared) light sources generate a reflection on the cornea whose position is of importance for subsequent calculations in the gaze tracking calculations. These light sources shine not only onto the eye but also into the eye, where they illuminate parts of the retina. If the light source is positioned very close to the optical axis of the camera, the light will be reflected back from the retina through the pupil, making the pupil appear bright (this is the same effect as the ‘red-eyes’ effect observed with small photo cameras where the flash light is close to the lens). Such systems are said to employ on-axis illumination, resulting in a bright-pupil effect as shown in Figure 2 (a). In contrast, systems positioning the light source at some distance from the camera employ ‘off-axis’ illumination, resulting in a dark pupil image.

The position of the pupil and of the glint(s) may be used for gaze estimation, and should, therefore, be determined accurately. However, several issues have to be considered if one is to achieve reasonable results. For instance, the reflections on the rim of the glasses in Figure 1 (b) should not be confused with the reflections on the cornea. The image in Figure 1 (a) shows usual artefacts such as visual noise and shining.

Figure 1. (a) Eye image from Li et al. (2005), as produced by common eye tracking systems. (b) Face image from image series used by Schmidt (2008).
Related Content

Novel System for Color Logo Recognition Using Optimization and Learning Based Relevance Feedback Technique
www.igi-global.com/article/novel-system-for-color-logo-recognition-using-optimization-and-learning-based-relevance-feedback-technique/195008?camid=4v1a

A Robust Color Watershed Transformation and Image Segmentation Defined on RGB Spherical Coordinates
www.igi-global.com/chapter/robust-color-watershed-transformation-image/73187?camid=4v1a

Watermarking of Digital Images With the Substitution of Low-Value Bits to Increase Capacity

Ensemble Classification System for Scientific Chart Recognition from PDF Files
www.igi-global.com/article/ensemble-classification-system-scientific-chart/75766?camid=4v1a