Basics of Structured Light

By Jorge Antón García, ECE '17

Introduction

How can you effectively go about tracking where an eye is looking at? Tracking an object is never a trivial task, and doing so when it does not emit anything is even more complex. One way to do so is to attach a strongly reflective marker or one that emits detectable waves (sound, magnetic field, light) on the object being tracked. Since this is not possible on an eye, a common approach is to use cameras. These can be used to track features, a collection of pixels within an image which are distinguishable. An example of a feature would be an edge, a corner, or a unique pattern of pixel colors. If these too are limited, then a pattern can be shined on the object. The reflection of this pattern on the object can be detected and used to simulate features. The generalization of using controlled signals to reflect off what is being tracked and back into the lens is called structured light. Throughout the paper we will discuss how structured light works and how it can theoretically be used in the realm of eye tracking as a way of providing more information about where the eye is looking.

Matching Lights

Tracking using structured light involves projecting a known pattern onto a surface, and given the pattern received by the cameras in the scene, being able to determine the pose of an object being tracked. The pose of an object is its relative position and orientation with respect to a reference point, which in many cases is a camera. To determine an object's position in space, one would first establish correspondence between the lights emitted and the lights detected. In other words, if three lights are shined at an object, all aligned horizontally, and three lights are detected, being able to deduce which light of the image corresponds to the LED on the left, middle, and right of the projector. The goal for matching these lights would be to be able to determine the relative pose between the camera and the object, as well as, between the object and the projector.

The simplest case of tracking by matching patterns would be to only project one light at a time. If the light was seen and it was the only light being projected, then the projected light is the detected light. However, to determine the pose of an object one would need to detect the light in five consecutive pictures (Livingston 47). If a camera takes a picture every 1/60 of a second, it would take 1/12 of a second to be able to take the five required pictures and determine the pose of the object. In this amount of time, if the eye were to move at the maximum speed, it could rotate up to 75 degrees which would make it difficult to estimate where the eye will be next (Visual Selection and Attention). It is also highly improbable that the eye remains completely still during a twelfth of a second. If a predictor-corrector algorithm were to be used and information from the eye's location in previous frames was incorporated, then 5 frames would only be required to calculate the pose initially or when tracking is lost. In all other cases, only one frame and the previous data are necessary. Due to blinking and the quick movements of the eye, it seems difficult to design a solution robust enough to not lose tracking.

Most approaches are application-specific and involve encoding the signal emitted to make each light unique. Examples of this would be changing the intensity or color of each light being emitted. For example, if one light is blue and another is red, then the camera can easily distinguish which light (the blue one or the red one) corresponds to the detected light. If there are a large number of lights it may be difficult to encode each by just varying these two parameters. This would also not be an effective approach if the background has a lot of colors and one cannot distinguish between the rebounded light within the image. Other options include exploiting the spatial neighborhood. Once one match is made, it can give you information about the points around it and where they could be. Patterns like lines, circles, cross-hairs, stripes, etc... could be used (Livingston 40). One could also incorporate the distances between points as well as which ones are neighbors when deducing correspondences. Difficulties arise when there are occlusions and some lights are covered. This would cause non-neighboring lights to appear as neighbors in an image.

It may be interesting in some cases of eye tracking, where the head can tilt, to create complete or partial rotational invariant patterns. This means that regardless of how the pattern is rotated it will always be unique. In partial rotational invariance, rotating the pattern 180 degrees creates a pattern identical to the one where there are no rotations. This is appropriate for eye tracking given that it is not possible to rotate one's head 180 degrees. Creating an approach which matches projected pixels to detected ones is essential in finding the offset from the camera to the eye.

Further Calibration

After matching the points, the direction from the center of the projector to the projected feature and the 2D location of the light on the image are known. To figure out the direction between camera and the place where the light hits the object, we must determine the intrinsic parameters of the camera through a calibration process. Some cameras come with a known intrinsic calibration, but for others one would need to test for their field of view, aspect ratio and focal length (Livingston 53). There are many resources online which talk about this procedure and many implementations like the ones found in open source libraries like OpenCV. The result of this calibration is a matrix with the camera's intrinsic values. When this is done, one can figure out the ray between the camera and each feature detected (OpenCV).

It is very difficult to know whether something in an image is a miniature version close up or a life-sized figure at a far distance. One option to solve this problem would be to place an object of known size in the image. Another option would be to use multiple cameras with known offset between them. When taking a picture and the object's position moves a lot between both camera pictures, then the object is close (MacCormick). This information, as well as using another method to get depth helps determine scale. This calibration is the final step to provide information about the angle between any projector, the eye, and the camera. Furthermore, with all the angles and one known distance, one could figure out the offsets between the camera and the feature (Livingston 50).

Conclusion

Where an eye is looking at is difficult to track because it does not emit anything. An effective way to locate gaze is by using a structured light approach. The benefits of this are that it provides distinct features to what otherwise with a pure optical approach may not be tracked. Knowing how the light bounces back from the eye gives one extra information that cannot be deduced just by tracking where the pupil is with a camera.

References

1. "Camera Calibration with OpenCV - OpenCV 2.4.13.2 Documentation" OpenCV. Web. 19 Apr. 2017

2. Livingston, Mark Alan. "Vision-based Tracking with Dynamic Structured Light for Video See-through Augmented Reality." (1998): n. pag. Web.+

3. Logan, Gavin. "1 Tracking Systems in VR." *Ppt Download*. N.p., n.d. Web. 20 Jan. 2017.

4. MacCormick, John. "How Does the Kinect Work?" (2006): n. pag. *Dickinson Selected Talks*. Web.

5. "The Parameters of Eye Movement." *The Parameters of Eye Movement*. N.p., n.d. Web. 20 Jan. 2017.

6. *Visual Selection and Attention*. Irvine: Donald Bren School of Information and Computer Sciences, n.d. PPT.