Foveated Rendering

Alejandro Morales and Liang Shi

Berkeley
UNIVERSITY OF CALIFORNIA

Background

The human visual system uses variable resolution sampling centered at the fovea to discard the vast majority of available visual information (1). This sampling scheme takes advantage of the high density of photoreceptors at the fovea to acquire high resolution information where the gaze is directed, while keeping more eccentric, or peripheral, visual information at a lower resolution. This creates a nearly scaling-invariant system that preserves semantic information about the object visualized as long as the focus is in the same place.

This variable sampling is illustrated in the sampling lattice shown in **Figure 1** from Van Essen and Anderson (1). Additionally, the formula below describes the relationship between eccentricity (E), defined as angular distance from the center of focus, and sampling interval (D). The slope (α) is estimated based on psychophysical data from the primate retina.

$$D = \delta + \alpha E = \alpha(E_0 + E) \approx 0.01(1.3 + E)^\circ$$

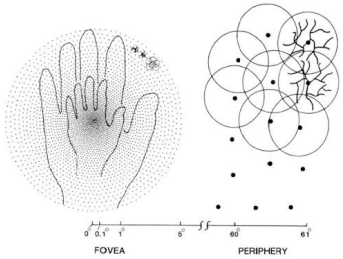


Figure 1. Eccentricity-dependent sampling lattice exhibited by the primate retina.

References

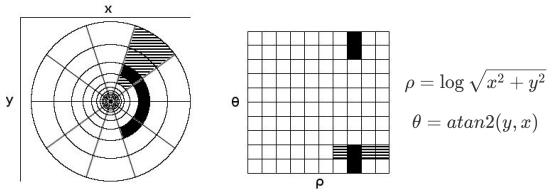
(1) David Van Essen and Charles H. Anderson. Information Processing Strategies and Pathways in the Primate Visual System. Van Essen and Anderson. 1995

Foveated Rendering Demo

Figure 2. Foveated rendering demo (https://alemorm.github.io/foveated-imaging/)

V1 Cortex Log-Polar Projection

A log-polar representation of the images, similar to that projected to the primary visual cortex (V1) from the retinal ganglion cells (RGC). The transformation was performed following the map shown in **Figure 3**.



$$\rho = \log\sqrt{x^2 + y^2}$$

$$\theta = atan2(y, x)$$

Figure 3. Log-Polar Transform from Cartesian coordinates

Implementation

The application was implemented in JavaScript, allowing users to upload an image and visualize the effect of foveated rendering with the fovea located at their cursor position. This effect was achieved real-time by applying low-pass filters to each pixel on the image, with the filter size dependent on the pixel distance from the fovea. We provide two different blurring methods, discrete and interpolated:

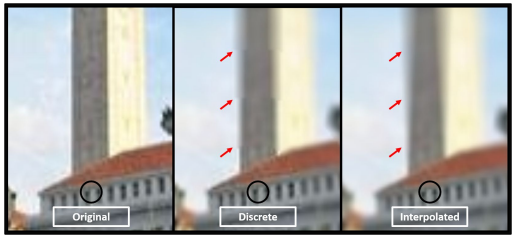


Figure 4. Original image with the types of interpolation. Red arrows show regions where different adjacent low-pass filters were applied. Black circles are the fovea location.

Iteratively applying low-pass filters to every pixel in an image is computationally demanding. This challenge becomes even greater as the image gets larger, due to the exponential increase in total pixels. To mitigate this issue, we compute a summed-area table (**Figure 5**) right after loading the image, reducing computation on each pixel to constant time and significantly accelerating the processing time.

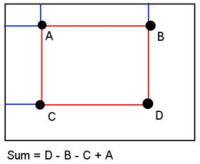


Figure 5. Computing pixel sum with summed-area table