Visualization of Large images across a Thinwire

(Thesis Proposal)

Ee-Chien Chang

January 12, 1998

1 Introduction

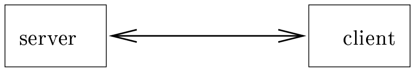


Figure 1: Thinwire client-server model.

We are interested in the visualization of large images across a network. This is based on the well known client-server model. The server holds some large images that the client wishes to visualize. As the network is of relatively low bandwidth, we call it a *thinwire* (Figure 1). A usual practice under this setting is progressive transmission [38] where a low resolution image is first sent across and it is then progressively and uniformly refined (Figure 2). A key observation is that, at any moment, the viewer is only interested in a region around a point (the *gaze point*) in the image. To exploit this, we propose a scheme where the viewer interactively indicates this region and the selected region has higher priority in the refinement process (Figure 3). As a result, the displayed image in the client-side does not has uniform resolution. We call such image a *space-variant* image [49, 46] and call the scheme *interactive progressive transmission.*

A fundamental difference between this approach and the usual progressive transmission approach is that we bring what the viewer wants to see into consideration, in other words, we place more emphasis on the visualization process. This additional element opens new perspectives on the problem. In this thesis, we focus on this difference. Three issues arise and the next three sections treat them in order.

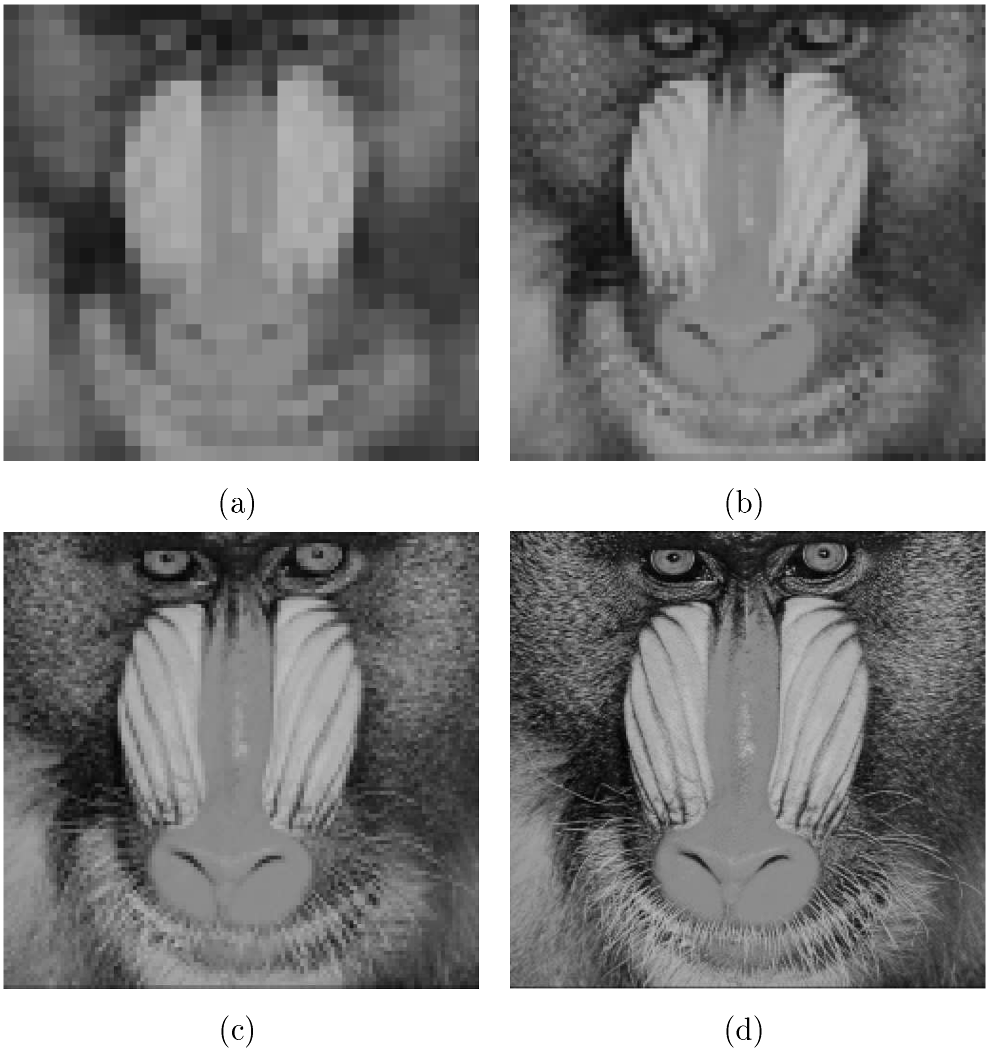
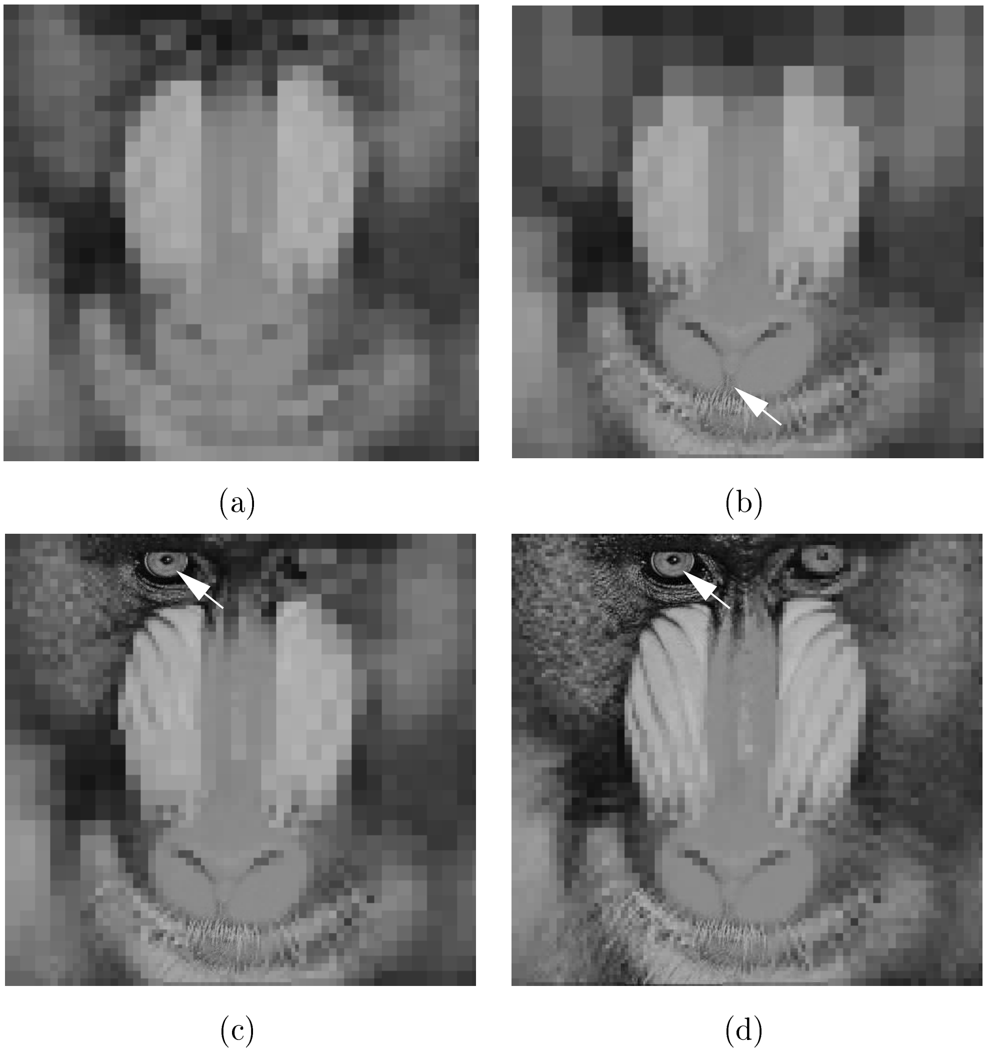


Figure 2: Progressive Transmission: A low resolution image (a) is first sent across. The subsequent image (b), (c) and (d) are refined progressively.



(a) (b) (c) (d)

Figure 3: Interactive progressive transmission: the white arrow points to the region selected by the viewer. Note that in (c), the details around the “mouth” retain. The position of the arrow remains unchanged from (c) to (d), and thus more refinements are performed on the “left eye”.

(a) Uniform resolution image. (b) Space variant resolution image

Figure 4: Foveation.

First, we need to understand what we mean by an image having varying resolution, and we need to know how to distribute the resolution on an image and how to progressively refine such a space-variant image. In section two, we look at a psychophysical finding of human vision and study space-variant image using two approaches. In the first approach, we propose a formulation that expresses the process of obtaining a space-variant image from a uniform image as a functional operator and study its operator matrix. In the second approach, we propose a scheme where, loosely speaking, the space-variant image is the one that minimizes a weighted distance from the original image, given a fixed budget of bits.

The second issue arises from the interactivity between the viewer and the system. If we treat what the viewer wants to view as a request for data, then, because of the thinwire restriction, it is necessary to reduce the actual amount of data requested from the server. This motivates the formulation of an on-line scheduling problem in section three. An interesting property of this problem is that a request could be scaled down and served at a lower level of service. We propose a few scheduling strategies.

The actual implementation of the system is not straightforward. Since the images are large, a more careful memory management strategy is required. Furthermore, as visualization is a real-time process, this further induces more constraints in the system design. In section four, we study these constraints in more detail.

2 Foveation

Figure 4(a) is a uniform resolution image whereas Figure 4(b) is a *foveated* image. We call the point of highest resolution the *fovea* and the process of going from a uniform image to a foveation image *foveation*. In general, we call an image with non-uniform resolution a *space-variant* image. These terms are borrowed from the psychophysical findings of biological vision system. Note that in Figure 4(b), resolution decreases gradually as the distance from the fovea increases. One of our aims here is to formulate this space-variant structure of an image.

2.1 Psychophysical Aspect of Foveation

It is well-known that our visual system has a space-variant nature where the resolution is high in the center of the visual field but gradually falling off towards the peripheral. Studies of this space-variant structure in the visual cortex could be traced back to [29, 16, 32], who suggest a well-defined map-like representation of visual field in the cortex. In the early 1940's, Talbot and Marshall [40] demonstrate and confirm this hypothesis. Subsequent studies by [47, 10], and mainly by Schwartz [33] give the well-known logmap mapping of the visual field in the visual cortex. Schwartz gives a recent survey on this topic in [34]. The logmap is characterized by two real constants K and β and can be conveniently expressed as a complex function $LOG : \mathbb{Z} \to \mathbb{Z}$ of the form:

$$LOG(z) := K \ln(z + \beta),$$

where a complex number $x + yi$ represents a point (x, y) in the Cartesian coordinate. Equivalently, the logmap maps the point (x, y) in the *visual field* to a point (ρ, θ) in the *visual cortex plane* where

$$\begin{aligned} \rho &:= K \ln\left(\sqrt{(|x| + \beta)^2 + y^2}\right), \quad \text{and} \\ \theta &:= \begin{cases} \tan^{-1}\left(\frac{x+\beta}{y}\right) & \text{if } x > 0 \\ -\tan^{-1}\left(\frac{x-\beta}{y}\right) & \text{otherwise} \end{cases}. \end{aligned} \tag{1}$$

This mapping is illustrated in Figure 5. We will consider different form of mappings from the visual field to the visual cortex plane, thus, in our context, the logmap is not necessary restricted to (1).

By discretizing the visual cortex plane uniformly, the value at each (ρ_i, θ_j) could be viewed as the result of applying an "averaging" operation (or low-pass filtering) on the neighborhood of (ρ_i, θ_j) in the visual cortex plane, which corresponds to a similar operation on the neighborhood of $LOG^{-1}(\rho_i, \theta_j)$ in the retina plan. Call the discrete points $\{(\rho_i, \theta_j)\}_{i,j \in \mathbb{N}}$ the *logmap* pixels. Note that the logmap determines the arrangement of logmap pixels in the retina plane. An advantage of using the logmap pixels to represent the visual field (rather than the uniform grid) is that it permits reduction of the total number of pixels while retaining high density of pixels in the "interesting" region, and this has motivated the design of many computer vision and visualization systems.

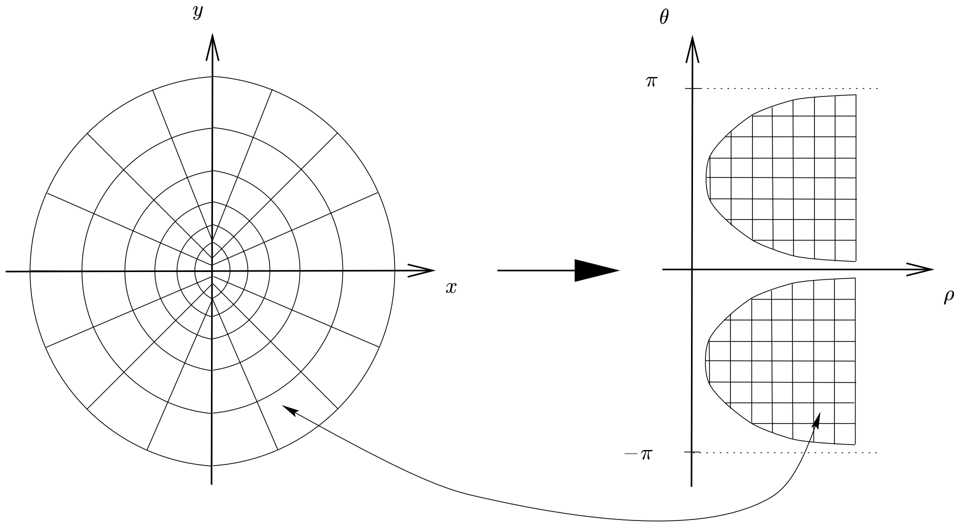


Figure 5: Transformation from the visual field to the visual cortex.

2.2 Applications of Foveation in Computer System

From the engineering point of view, it is not necessary to adhere to the exact form of space-variant mapping obtained from psychophysical findings. For example, Panerai et. al. [13] use a logmap of the form,

$$\begin{aligned} \rho &:= K\ln(\sqrt{x^2+y^2}+\beta), \quad \text{and} \\ \theta &:= \begin{cases} \tan^{-1}\left(\frac{x}{y}\right) & \text{if } \ x>0 \\ -\tan^{-1}\left(\frac{x}{y}\right) & \text{otherwise} \end{cases}. \end{aligned} \tag{2}$$

They also study various arrangements of logmap pixels (in the retina plane), namely *log-Cartesian* and *horizontal log-Cartesian* as illustrated in Figure 6. Another interesting arrangement is the one obtained by blending two arrangements: the usual uniform arrangement within the fovea and the space-variant arrangement on the peripheral. Examples of this configuration are described in [31]. A disadvantage of this configuration is the difficulty in merging the two arrangements smoothly.

The values at the logmap pixels can be obtained from a scene using a foveated vision chip (or seeing silicon) where the photoreceptors are organized in a similar way as the biological retina. Each photoreceptor carries out a sampling process by a sampling function usually of the form of a Mexican hat, whose width depends on the location of the photoreceptor. The width of a sampling function increases linearly as the distant of the photoreceptor from the center of the chip increases. Therefore, the process is a non-uniform sampling process. A survey on some of the past and current technology of vision chip could be found in [31],

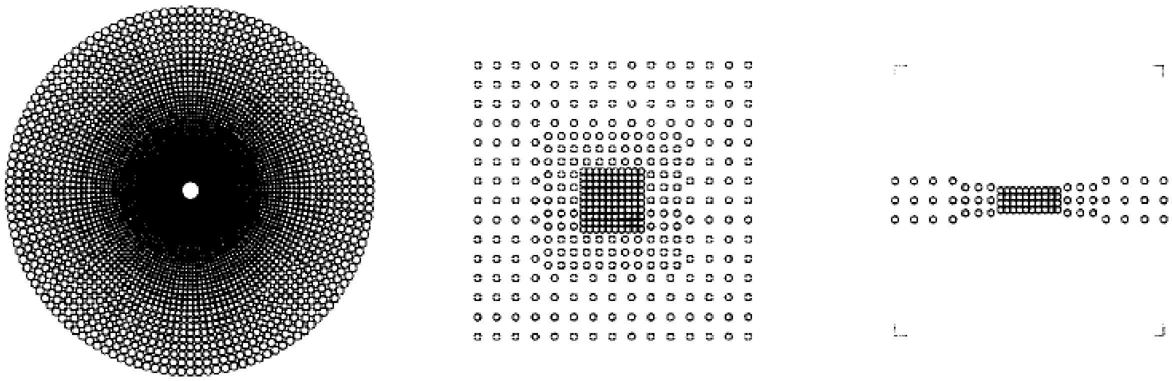


Figure 6: The logmap arrangement (given by (2)), log-Cartesian and horizontal log-Cartesian arrangements. Image reproduced from [13].

which is also available in the web [30].

Another source to obtain the value at the logmap pixels is from a two dimensional uniform resolution raster image, for example Figure 4 (a). One approach to extract the values is by directly "emulating" the non-uniform sampling process, that is, the retina plane is first partitioned into blocks where each block corresponds to a logmap pixel, and then the value of each logmap pixel is obtained by averaging all the original (uniform) pixels in the same block. The real-time system designed by Kortum [18] and the real-time system CORTEX-I developed by Wallace et. al. [46] use this approach. The partition is usually precomputed and stored in a lookup table, and thus, the "emulating" process is fast but lack flexibility in the sense that the arrangement of the logmap pixels is fixed. A more flexible approach is to first compute a hierarchical representation of the image, and the foveated image is obtained by cut-and-paste from different scales of the image. The method proposed by Burt [8, 2, 7] which uses Gaussian pyramid as the hierarchical representation of the image belongs to this approach.

Machine Vision A uniform image contains too much information and this is always a bottleneck in real-time vision systems. Two straightforward ways to reduce information are by lowering the resolution uniformly or by reducing the size of the visual field. Foveation provides a simple and fast blending of both means of reduction. Yeshurun and Schwartz [49] used the term *space-variant vision* to refer to machine vision using foveation. The performance of a space-variant vision system could be further enhanced by dynamically moving the fovea so as to gather relevant information. This methodology is also known as *smart-sensing* introduced by Burt [7]. Again, this dynamic nature is motivated by the human vision. An implementation of such a system is CORTEX-I[46, 4].

Another interesting property of foveation lies in the geometry of the logmap. Note that rotation and scaling in the retina plane amount to translation along the θ and ρ axis in the visual cortex plane respectively. Apparently, this is a desirable property in some vision processes, for example, corner detection. In addition, based on experimental results, Panerai et. al. [13] report that computation in the visual cortex plane is more robust in finding correlation of two images I_{left} and I_{right} where both images are obtained from a binocular camera system.

A lot of basic image processing operations depend on the neighborhood relationship of pixels. In the usual uniform image, this neighborhood is usually taken as the 4-neighbor or 8-neighbor depending on whether the four corners are to be included. It is not clear how the neighborhood relationship should be defined for the logmap pixels. Wallace et. al. [46] study varies configuration of connectivity graph for the logmap pixels, where the connectivity graph represents the neighborhood relationship of the logmap pixels: two pixels are adjacent if and only if there is an edge between them.

Visualization The existence of a space-variant nature in our visual system suggests that a foveated image and the full resolution image is visually indistinguishable if the viewer's gaze point is fixed at the fovea. This observation could be confirmed by psychological experiment and has been exploited by many visualization systems.

Since the landmark NSF Report in 1987 [27], visualization is an active research area. Visualization is the process whereby the viewer gains insight of a large data set. A main concern in visualization is the extraction of information from a large data set and the subsequent transformation of this information to some geometric objects. These geometric objects are then presented to the viewer through the graphics engine. However, the number of these geometric objects may be large, and the issue of how to visualize this large geometric data set is also interesting. A possible approach is to allow the viewer walking through these objects interactively. Of course, this method is computational intensive due to the size of the data and the real-time requirement. This is where foveation could play an active role.

Volume rendering, which could be viewed as a projection from a three dimensional data to a two dimension image, is inherently computational intensive due to the vast data size. In three dimension, foveation results in even more significant data reduction, which in turn, leads to reduction of rendering time. Since volume data is a generalization of image data, techniques on image foveation could be extend to volume naturally, except that it poses more computational challenge. Levoy and Whitaker build a system that achieve real-time rending of volume data [19] using foveation. They perform volume rendering at different resolution and the foveated image is obtained by pasting images with different resolution using interpolation. Other examples are flight simulation systems designed by [42, 12].

Recently, there are growing interests in visualizing large data sets across the Internet, and in these applications, the low transmission rate of the network is a bottleneck. Foveation, in this case, helps to reduce the network workload. As opposes to the previous application, the date here could be the simple array data, for example image or video.

Using foveation or region of interests (ROI) to reduce network workload is an active area in video conferencing [41, 15, 3, 1]. Attention is also focused on the means to select ROI, for example in [41, 11]. Note that Video and still images are very different in nature and requirement. The video transmission stressed even more on real-time performance, but there is less requirement of progressive refinement since every frame or image displayed is not the same. For still image, it is well known that allowing the user to specify the area of interest could greatly improve compression rate. For example, the commercial Summus' Wavelet Image (WI) has the feature of multiple "region of interests focusing" [39]. Here, the role of the user is to select the region of interests before the image is compressed and the selected regions are compressed with less distortion. On the other end, the role of the viewer is passive in the sense that he is not allowed to choose the region of interests.

A point to note here is that the viewer's gaze point is not fixed, in fact, it is actively moving on the image. Thus, to provide a foveated image that is indistinguishable to the viewer throughout the visualization process, it is necessary to track the viewer's gaze point and produce the corresponding foveated image. Example of systems using eye tracking device are [19, 18].

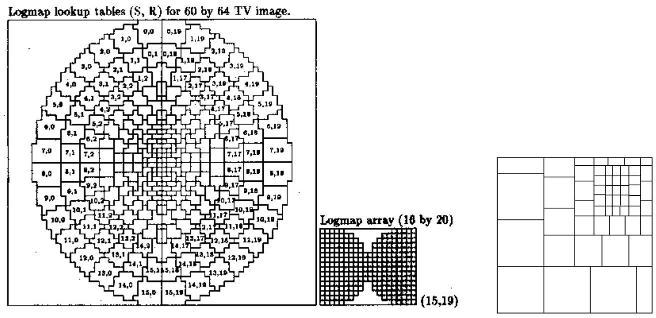
We would like to clarify that in our thinwire application, we do not intend to track the viewer's gaze point. Instead, the viewer is to "cooperate" with the system by selecting his gaze point using the conventional pointing devices, for example mouse, joystick or even keybroad. We also do not aim to use the logmap to produce foveated images that are indistinguishable from their corresponding full resolution images. Instead, we use the logmap as a guide to the "distribution" of resolution on the images.

2.3 Foveation Operator

In this section, we propose a formulation of foveation motivated by the logmap.

2.3.1 Definition in one dimension.

We begin with one dimensional functions. A *foveation* of a function $f : \mathbb{R} \to \mathbb{R}$ is determined by a *weight function* $w : \mathbb{R} \to \mathbb{R}_{\geq 0}$ and a *scaling function* $g : \mathbb{R} \to \mathbb{R}$. The weight function



(a) (b)

Figure 7: Foveation as a non-uniform sampling process. Figure (a) is reproduced from [46] and (b) is reproduced from [18]

is non-negative and $w(x) = 0$ only for finitely many x. The scaling function g is integrable and normalized so that $\int_{-\infty}^{\infty} |g(x)|dx = 1$.

Define the *foveation* operator T^{fov} as

$$\left(T^{\text{fov}} f\right)(x) := \int_{-\infty}^{\infty} f(t)\frac{1}{w(x)}g\left(\frac{t-x}{w(x)}\right) dt. \tag{3}$$

By writing

$$g_x(t) := \frac{1}{w(x)}g\left(\frac{t-x}{w(x)}\right), \tag{4}$$

T^{fov} can be rewritten as

$$\left(T^{\text{fov}} f\right)(x) = \langle f, g_x \rangle.$$

Call the result of the above operation a *space-variant image.*

Treating T^{fov} as an integral operator of the form

$$T(f)(x) := \int K(x,y)f(y)dy, \tag{5}$$

then in our case, the kernel $K(x, y)$ is the scaling function $g_x(y)$.

2.3.2 Definition in two dimension.

In two dimension, the foveation is characterized by a two-dimensional weight function $w : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_{\geq 0}$, and a two-dimensional scaling function $g : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. The scaling function is integrable and normalized as follow:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |g(s,t)|\, dsdt = 1.$$

The weight function is non-negative and vanish only for finitely many (x, y). Let

$$g_{x,y}(s,t) := \frac{1}{w(x,y)^2} g\left(\frac{s-x}{w(x,y)}, \frac{t-y}{w(x,y)}\right).$$

Define the two dimensional foveation as

$$\left(T^{\mathrm{fov}} I\right)(x,y) := \langle I, g_{x,y}\rangle. \tag{6}$$

One way to obtain a two-dimensional weight function is by extending a one-dimensional weight function $w(.)$ in the following way:

$$w(x,y) := w(\|(x,y)\|_2),$$

where $\|.\|_2$ is the usual 2-norm.

We are interested in a special class of weight functions. A *standard weight function* is one of the form

$$w_{\mathrm{std}}(t) := \alpha|t-\gamma| + \beta, \tag{7}$$

where α, β and γ are fixed constants and $\alpha, \beta \geq 0$. We call α the *rate*, β the *foveal resolution* and γ the *gaze point*, and call the resulting space-variant image a *foveated image*. Figure 4 (b) is the result of applying T^{fov} on Figure 4(a) with a standard weight function.

We could obtain new weight function by combining several weight functions. Given two weight functions, their *blended weight function* w is defined as

$$w(x,y) = \min\{w_1(x,y), w_2(x,y)\}.$$

We call a weight function composed by finitely many weight functions a *multi-fovea* weight function and the corresponding space-variant image a *multi-foveated image.*

We only consider standard weight functions and multi-fovea functions. Beside the physiological motivation, a multi-fovea weight function is easy to manipulate since it is composed by finitely many standard weight functions where each in turn can be easily represented by their rate α, foveal resolution β and gaze point γ. This simple description is desirable in the thinwire application.

2.3.3 Motivations by logmap.

The definition of T^{fov} and standard weight function is motivated by the logmap. We would like to capture the situation where sampling is performed in the visual cortex plane. A disadvantage of consider uniform sampling in the visual cortex plane is that the corresponding

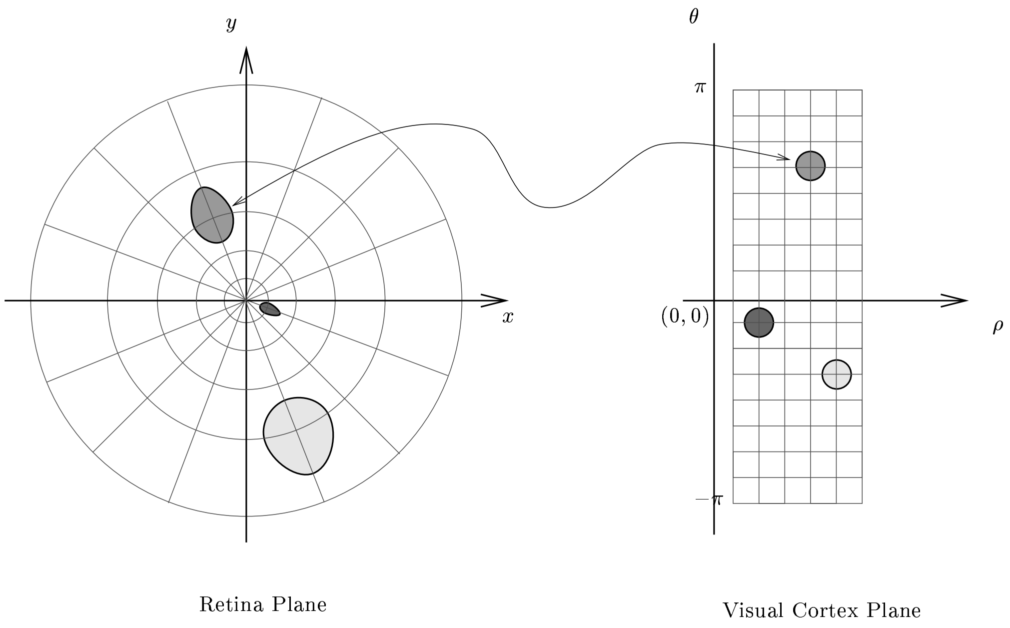


Figure 8: The corresponding neighborhoods in the visual cortex plane and the retina plane. Note that the neighborhoods in the retina plane have different orientation and shape.

sampling functions in the retina plane have different "orientation" and "shape", depending on their locations (Figure 8). This inconsistency may cause difficulties in computation. Instead, we would like to keep the orientation and shape of the sampling functions but varies their width according to their locations (Figure 9).

Note that in the retina plane, the width of the sampling function grows linearly as its distance from the fovea increases. This observation leads to the definition of standard weight function. In one dimension, if the weight function is $w(x) = \alpha|x| + \beta$, then

$$(T^{\text{fov}} f)(x) = \int_0^\infty f(L^{-1}(t))g(L^{-1}(t-x))dt,$$

where $L(t) := sign(t) \log(\alpha|t|+\beta)$. The values of $(T^{\text{fov}} I)(x)$ at a non-uniform grid $x_i := L^{-1}(i)$ for some constant a, c and $i \in \mathbb{N}$, could be treated as the value of the logmap pixels.

In two dimension, if

$$w(x, y) := \sqrt{r(r+1)},$$

where $r := \sqrt{x^2+y^2}$, then

$$(T^{\text{fov}} I)(x, y) = \int_{-\pi}^{\pi} \int_c^\infty I(x, y)g(x, y)dRd\theta,$$

where c is some constant and

$$R \;\; := \;\; \exp(r+1)$$

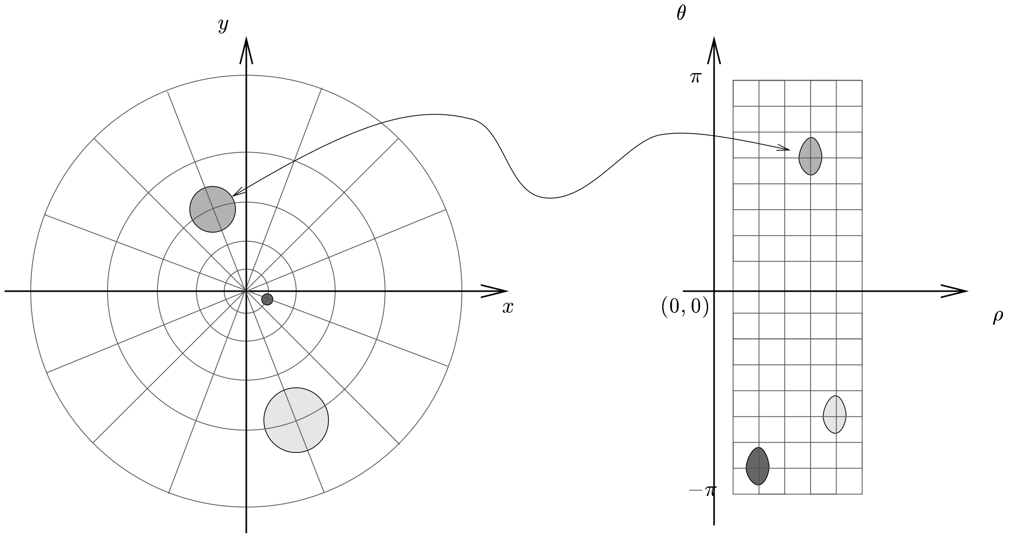


Figure 9: As opposed to Figure 8, the neighborhoods in the retina plane have same orientation and shape.

$$\theta \;:=\; \begin{cases} \tan^{-1}\left(\frac{x}{y}\right) & \text{if } \; x > 0 \\ -\tan^{-1}\left(\frac{x}{y}\right) & \text{otherwise} \end{cases} .$$

2.3.4 Remark.

1. Consider an integral operator (5). If the function f is sampled at scale N^{-1} as a N-vector and the kernel as an $N \times N$ matrix K, then a direct method to compute $T(f)$ takes $\theta(N^2)$ arithmetic operations. However, in some cases, we could do better. For example, if the matrix K is diagonal, then we have a trivial linear time algorithm.

   One way to analyze an operator T is by studying its operator matrix A whose entries

   $$a_{i,j} := \langle T(g_i), g_j \rangle,$$

   where $\{g_m\}_{m\in\mathbb{N}}$ is an orthonormal base. We are interested in the case where $\{g_m\}_{m\in\mathbb{N}}$ is a orthonormal wavelet base. For example, the compactly supported wavelet constructed by Daubechies [9] following the earlier work of Meyer [28] and Mallet [21].

   Similar to image compression, one could compress this matrix by suppressing the entries below a threshold value, and the compressed matrix is a good approximation of the original T. For many operators, by choosing an appropriate wavelet, the compressed matrix $\tilde{A}$ could be sparse and has very thin band, that is, $|i-j| > k \Rightarrow \tilde{a}_{i,j} = 0$, where k is some small constant. A well-known operator that can be compressed in this way is the Calderón-Zygmund operator. We refer the readers to the work of Beylkin, Coifman and Rokhlin [5, 6].

2. It is interesting to know whether the foveation operator is bounded. An operator T is bounded if there is a constant B such that for any function $f \in L^2(\mathbb{R})$,

$$\|T(f)\|_2 \leq B\|f\|_2.$$

It is easy to construct an example of w and g such that the corresponding T^{fov} is not bounded. Let

$$f_1(x) := \begin{cases} x^{1/3} & \text{if } |x| < 1 \\ 0 & \text{otherwise} \end{cases},$$

which is in L^2. If the scaling function is

$$g(x) := \begin{cases} \frac{1}{2} & \text{if } |x| < 1 \\ 0 & \text{otherwise} \end{cases},$$

and the weight function is

$$w(x) := \frac{1}{x},$$

then $T^{\text{fov}} f_1$ is of the order $\theta(x^{-1})$ near the origin and thus it is not in L^2.

An approach to prove the boundedness of an operator is by studying its operator matrix. If the matrix is bounded, then so is the operator. A simple but yet useful lemma to show the boundedness of a matrix is the Schur's lemma.

Lemma 1 *(Schur). Let T be an operator whose matrix elements in an orthonormal base $\{g_n\}_{n\in N}$ are $\theta_{n,m} = \langle Tg_n, g_m\rangle$. If there are two sequences of positive number $\{w_m\}$ and $\{\hat{w}_m\}$ and a constant B such that*

$$\sum_{m=0}^{\infty} |\theta_{n,m} w_m| < B\hat{w}_n \tag{8}$$

and

$$\sum_{n=0}^{\infty} |\theta_{n,m} \hat{w}_n| < Bw_m, \tag{9}$$

then

$$\|T\|_2 \leq B.$$

Examples of this approach could be found in [14, 22].

2.4 Progressive Transmission

The previous formulation treats foveation as a non-uniform sampling process. It has a few limitations in the interactive progressive transmission setting. By the definition of T^{fov}, the original function is a "poor" approximation to a foveated image. Imagine a situation in our thinwire application, where a client requests for a foveated image but the server manages to provide the original image I. By using the mean square error $\|T^{\text{fov}}(I) - I\|_2$ as a measure of error, the performance of this server is "poor". However, on the contrary, the original image is very much preferred by the client.

To overcome this limitation, we examine the progressive transmission scheme in general. Most schemes use the mean square error as the measurement of error which is measured on the retina plane. Following the same spirit as in the definition of foveation operator, we consider the mean square error in the visual cortex plane, which corresponds to a weighted mean square error in the retina plane. A foveated image in this setting, loosely speaking, is the image that minimizes its weighted distance from the original image but represented by a fixed (predefined) number of bits.

2.4.1 Image compression and Progressive Transmission.

Progressive transmission was introduced by Sloan and Tanimoto [38]. Since then, it is an active research area and most current image compression schemes include feature of progressive transmission. The popular graphic format GIF and JPEG [45] already have progressive transmission mode. A survey on progressive transmission could be found in [44].

Under a progressive transmission scheme, an image I is first transformed to a code and stored as a linear bit stream $b_0, b_1, \ldots, b_n$. For any t, it is possible to reconstruct an approximation I_t of the original I based on $b_0, b_1, \ldots, b_t$. It is desirable that the distortion is bounded by ϵ_t, that is

$$\|I_t - I\|_2 < \epsilon_t, \tag{10}$$

and ϵ_t has fast decay as t increases.

One would argue that the mean square error in (10) may not be the right measure. It is well known that the sensitivity of the human vision varies across different frequencies [25]. However, mean square error is easy to manipulate mathematically. If other measures are preferred, usually we could approximate the desired measure by a weighted mean square error (which we will describe later). Many compression schemes, for example JPEG, use the weighted mean square error as a guide for compression.

Any lossy image compressive scheme naturally suggests a possible progressive transmission scheme, which could be obtained by progressively improving the quality of compression. Most lossy image compression schemes could be described in the framework of transform code. Under this framework, an image is encoded in three steps: transformation, quantization and lossless compression.

Transformation. This first step applies an invertible linear transformation to the image, in other words, it decomposes the image with respect to a base $\mathcal{B}$. There are many types of transformation. To name a few, the Karhunen-Loeve transformation (also known as principal orthogonal decomposition), which is optimal in the sense that it "completely" decorrelates the image by diagonalizing the covariance matrix; the DCT (Discrete Cosine Transform) which is used by JPEG; and subband coding. A subband coding is determined by a pair of filters, a low-pass filter H and a high-pass filter G. Its transform code is obtained by repetitive application of convolution by H and G follows by decimation on the image. Thus, the transform code is a hierarchical representation of the image. The fast wavelet transform, Gaussian and Laplican pyramid could be viewed as a form of subband coding.

It is desirable that the base $\mathcal{B}$ is orthonormal with respect to the 2-norm $\|.\|_2$. If this is the case, then the transformation has the energy invariant properties which implies that an error ϵ occurs in a coefficient induces a same error ϵ in the represented image. This ability of predicting error is very useful in analyzing and design compression scheme. Furthermore, orthonormal base tends to decorrelate the image better.

Quantization. After transformation, the coefficients $\{a_m\}_{m=1}^{M}$ are then *quantized*, that is, each coefficient a_m is approximated by $\tilde{a}_m$ taken from a finite set of values. The quantization error is the error $a_m - \tilde{a}_m$. This steps contributes to the lossy feature of the transform code.

A quantizer Q is a staircase function characterized by $x_i, i = 0, \ldots, N-1$ and $b_i, i = 0, \ldots, N$:

$$Q(x) = x_i \;\text{ if }\; x \in [b_i, b_{i+1}).$$

Call each $[b_i, b_{i+1})$ a bin. In all quantizers considered in this report, $x_i := \frac{1}{2}(b_i + b_{i+1})$. In the case where all bins have equal size, we call it the *uniform quantizer*. We are only interested in scaler quantization where each $\tilde{a}_m$ is determined solely by the value of a_m, as opposes to the vector quantization where $(\tilde{a}_{m_1}, \tilde{a}_{m_2}, \ldots, \tilde{a}_{m_k})$ are jointly determined by the vector $(a_{m_1}, a_{m_2}, \ldots, a_{m_k})$ for some k.

For a random source A and a quantizer Q, let $\tilde{A} := Q(A)$. If the number of bins is fixed, say N, to minimize the expected mean square error $E\{(A - Q(A))^2\}$, the Lloyd-Max algorithm [26, 17] gives the optimal solution where smaller bins are "allocated" to more popular values. However, if we fix the expected mean square error, say D, and would like to minimize the entropy of $\tilde{A}$, surprisingly, the uniform quantizer outperforms the optimal quantizer obtained by Lloyd-Max algorithm; in fact it is near optimal for small D (or large number of bins) assuming A is well-behaved (for example, when A is Laplacian [48] or the quantizer and A satisfy the high bit rate assumption described in [23]). Note that the entropy of $\tilde{A}$ is a lower bound for the average bit size needed to code $\tilde{A}$ [35].

In the context of progressive transmission, there are two types of scaler quantization: multiscale and embedded quantization. In multiscale quantization, the data is first quantized by a coarse quantizer and the quantization error is then quantized by a finer quantizer. In the embedded quantization [43], the output of the coarse quantizer coincide with the output of the finer quantizer applied in the next step. As a result, for each refinement, the refined quantized values can be obtained by concatenation of the previous values with the additional bits. On the other hand, for each refinement, in the case of multiscale quantization, an arithmetic addition is required for each coefficient. Therefore, embedded quantization is computationally more efficient. Furthermore it is also more efficient in achieving low distortion [43].

Lossless Compression. This step is also known as *channel compression*. After quantization, the image is represented as a sequence of bytes. Lossless compression applies an invertible transformation to remove redundancy. A major framework for lossless compression scheme is the entropy encoding. Let A to be a random source which takes value among a finite set of symbols $\{a_i\}_{1\leq i\leq k}$. An entropy coding scheme determines a binary coding for the symbols so as to minimize the expected bit size. Let

$$p_i := \mathrm{Prob}(A = a_i).$$

The entropy of A is defined as

$$-\sum_{i=1}^{k} p_i \log p_i.$$

The Shannon theorem shows that the entropy is a lower bound for the expected bit size to encode A. Popular entropy compression schemes are Huffman coding and arithmetic coding.

Many efficient schemes in fact are not based on entropy coding. Perhaps this is because entropy coding assumes that the A_i's are independent, which in fact is not the case for most data sets. An example is the simple run-length code that encode a consecutive sequence of zeros by the number of zeros. JPEG employs run-length coding followed by an entropy coding. Consider the wavelet coefficients of an image. By visual inspection, one could conclude that there is coherence across the scales. An example that exploit this observation is the zero-tree developed by Shapiro [36].

Using transform code to generate foveated images. Most current image compression schemes are concerned with uniform progressive transmission, although generalization to space-variant progressive transmission is not difficult; this is especially so in the cases of wavelet transformed code, since wavelet is space localized. For JPEG, since an image is first subdivided into non-overlapped blocks of 8×8 sub-image, and each block is treated independently, they can be easily modified to support the space variant progressive transmission. So, the question now is, given a standard weight function, how should the quantization be done in order to obtain the corresponding foveated image.

2.4.2 Bits Allocation of Transform Code.

We now treats the three steps of transform code more formally but restrict the discussion on the cases where the base $\mathcal{B}$ is orthonormal, and where scaler quantize and entropy compression is applied in the last two steps.

We follow the approach in [23]. Let $\mathcal{B} = \{g_m\}_{0 \leq m < N}$ be an orthonormal basis. A function/image f is represented by its coefficients $\{a_m\}$ with respect to $\mathcal{B}$ where each $a_m = \langle f, g_m \rangle$.

Let Y, $\{A_m\}_{0 \leq m < N}$ and $\{\tilde{A}_m\}_{0 \leq m < N}$ to be the random variables of of the signals/images f to be coded, the coefficient $\{a_m\}_{0 \leq m < N}$ and the quantized $\{\tilde{a}_m\}_{0 \leq m < N}$ respectively. Thus

$$Y = \sum_{m=0}^{N-1} A_m g_m,$$

and each

$$A_m = \langle Y, g_m \rangle.$$

Let $\tilde{Y}$ to be the random variable of the represented function (after quantization), which is:

$$\tilde{Y} := \sum_{m=0}^{N-1} \tilde{A}_m g_m.$$

Since g_m is orthonormal, the expected mean square error of approximating Y is

$$E\{\|Y - \tilde{Y}\|_2^2\} = \sum_{m=0}^{N-1} E\{|A_m - \tilde{A}_m|^2\} \|g_m\|_2^2 = E\{|A_m - \tilde{A}_m|^2\}. \tag{11}$$

Let r_m be the entropy of $\tilde{A}_m$. Consider the following problem.

(a). Given a random source A_m and a fixed constant D, what is the quantizer such that the expected mean square error is D and the entropy of $\tilde{A}_{m_0}$ is minimized.

Let

$$r := \sum_{m=0}^{N-1} r_m.$$

Note that r is the entropy of the joint distribution of $\{\tilde{A}_m\}$ only if all $\tilde{A}_m$ are independent. In general, r may be larger. As observed before, we are only interested in scaler quantization, thus we still use r to measure the average bit size. The question (a) is further extended to the whole function/image:

(b). Given a random image Y and a base $\mathcal{B}$, a fixed constant D, what is the quantizer such that the expected mean square error is D and r is minimized.

It can be shown that if the base induces equal distortion in each direction of g_m, then using uniform quantizer for all coefficients is optimal. This result can be extended to a weighted case where the expected mean square error (11) is replaced by a weighted mean square error,

$$\sum_{m=0}^{N_1} \frac{1}{w_m^2} E\{|A_m - \tilde{A}_m|^2\}. \tag{12}$$

In this case, the bin size for A_m is $\triangle w_m$, where $\triangle$ is a constant depends on the expected weighted error D.

2.4.3 Weighted progressive transmission.

As mention before, for progressive transmission of a foveated image, we propose to replace the means square error in (10) by a weighted norm $||.||_w$. Given a weight function $w : \mathbb{R} \to \mathbb{R}_{>0}$. Recall that a weight function $w(x) \geq 0$ for all x and $w(x) = 0$ only for finitely many x. For two one-dimensional functions $f : \mathbb{R} \to \mathbb{R}$, $g : \mathbb{R} \to \mathbb{R}$, and a weight function w,

$$\langle f, g \rangle_w := \int \frac{f(x)g(x)}{w(x)} dx.$$

Define the weighted norm of f as

$$||f||_w := \sqrt{\langle f, f \rangle_w}. \tag{13}$$

Consider (11), note that the original basis is not longer orthonormal under this norm and the subsequent arguments would not hold. However, it seems that for a standard weight function, under certain conditions we could approximate

$$||f||_w^2 \approx \sum_i \langle f, g_i \rangle ||g_m||_2^2. \tag{14}$$

If this approximation holds, then we can use (12) to approximate the weighted norm (13). where

$$w_m := \frac{1}{\|g_m\|_2}.$$

We need to find the conditions for the approximation (14), and to determine the value of w_m.

3 On-line Scheduling with Level of Service

Scheduling is an important issue in resource management. Suppose I is an instance of a scheduling problem, usually a collection of jobs or requests, the goal is to find an optimal *schedule* which achieves minimum cost or maximum merit points gained.

We are interested in on-line scheduling. In this setting, the problem instance is released gradually, a single job or request at a time, whereas the scheduler upon receipt of a new job/request, bases on the partial information available, is required to make decision and output a part of the schedule. Performance of an on-line scheduler could be measured by its competitiveness, introduced by Manasse, MacGeoch, and Sleator [24] and in the earlier work by Sleator and Tarjan [37]. Suppose S is a scheduler and $S(I)$ is the schedule where I is the instance, then S is c-competitive if for any instance I,

$$\mathit{merit}(\texttt{opt}(I)) \leq c \ . \ \mathit{merit}(S(I)) + b,$$

where b is some constant and $\texttt{opt}(I)$ is the optimal off-line schedule. The *competitive ratio* of a scheduler S is defined by

$$C := \sup_I \frac{\mathit{merit}(\texttt{opt}(I))}{\mathit{merit}(S(I))}.$$

We say that a problem is *no better than* c-competitive if there exist no scheduler that is c_1-competitive, for any $c_1 < c$. A classic example is the paging problem where the main memory can only holds at most k pages and the goal is to minimize the number of page faults. Sleator and Tarjan [37] show that the LIFO (last in first out) and LRU (least recently used) strategies are k-competitive and the problems is no better than k-competitive. The paging problem is a special case of the k-server problem introduced by Manasse, MacGeoch, and Sleator [24].

There are a few variants on the computational models. One may consider randomized scheduler. A randomized scheduler S, as opposed to the deterministic scheduler, is allowed to toss coins randomly. Given an instance I, let $S(I)$ to be the random variable of the schedule output by S. We say that S is c-competitive if for any I,

$$merit(\texttt{opt}(I)) \leq c \ . \ E[merit(S(I))] + b,$$

where b is a constant. Similarly, we say that S has competition ratio C if

$$C := \sup_{I} \frac{merit(\texttt{opt}(I))}{E[merit(S(I))]}.$$

Average case analysis is another major branch on the analysis of on-line problem. Here, the instance is drawn from a prior known distribution, and a scheduler S is measure by its expected performance. If we write I to be the random variable for the instance, then the expected performance is

$$E[merit(S(I))].$$

While the k-server and the paging problem focus on the fundamental issue of how to "distribute" jobs among k servers, the case where there is only one server is also interesting. An example is the on-line intervals packing problem where the instance consists of open intervals and a schedule is a subset of non-overlapping intervals. Lipton and Tomkins [20] study a variant where the intervals are restricted to two types: intervals with unit size and intervals with size k. The goal is to cover the real line as much as possible. They give a randomized scheduler that is 2-competitive.

We propose an on-line scheduling problem motivated by the thinwire model. If we treat what the viewer wants to view as a request for data, then, throughout the visualization process, more requests will be generated than the server could serve. Thus, it is necessary to select the more importance requests which are to be sent to the server. An interesting property of this problem which we exploit is that: a request could be scaled down and served at a lower level of service.

3.1 On-line scheduling with Level of Service

In this user-manager setting, the user issues requests and the manager tries to serve the requests with its limited resource. An instance is a collection of requests. Each request q has four parameters, (s, t, u, v), where s, the *start time*, is the time the request is issued; t, the *deadline*, is the time the request has to be completed; u is the size of the request and v is the *weight* of the request. Write $st(q), dl(q), sz(q), wt(q)$ for its start time, deadline, size and weight respectively. The requests are issued one at a time and if a request p is issued before q, then $st(p) \leq st(q)$. The manager can serve any issued request but at any moment, he can only serve one request. A request q is completely served if it has been served for a total of $sz(q)$ time. A request can not be served if it's deadline has already passed or it has been completely served. The service is *preemptive* without penalty, that is, the manager could switch from one request to another request at anytime and he could also switch back to an uncompleted service. However, the number of such switchings must be finite. For each completely served request, the manager gains $sz(q)wt(q)$ merit points. Unlike usual scheduling problem, a partially served request can still contribute a same proportional amount of merit points. If a request q is served for a total of x time, then the manager gains $xwt(q)$ merit points. The goal of the manager is to maximize the total merit points gained.

Call a request q *alive* at time t_0 if $st(q) \leq t_0 \leq dl(q)$. With respect to a manager, at time t_0, some requests could have been partially or completely served within $(-\infty, t_0]$. Call those alive requests that are not yet completely served the *pending* requests at time t_0.

3.2 Examples of Manager

Here are two examples of manager, `Greedy` and `EndFit`. At any one time, `Greedy` always serves the current heaviest pending request, whereas `OptFit` always serves according to the off-line optimal schedule of the current pending requests.

Lemma 2 *The competitive ratio of* `Greedy` *is at least 2.*

Proof. Take an instance I_1 which contains two requests $q_1 := (0, 2, 1, 1+\epsilon)$ and $q_2 := (0, 1, 1, 1)$, where ϵ is some positive constant. (Figure 10). Thus,

$$\frac{merit(\texttt{opt}(I_1))}{merit(\texttt{Greedy}(I_1))} = 2 - \delta,$$

for some $\delta > 0$. Note that $\delta \to 0$ as $\epsilon \to 0$. Thus, we have the claimed result. **Q.E.D.**

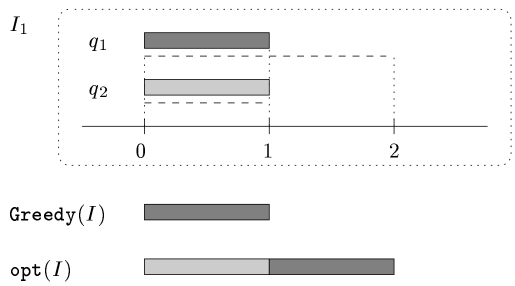


Figure 10: This example demonstrates that the competitive ratio of `Greedy` is at least 2.

Lemma 3 *The competitive ratio of* `EndFit` *is at least 2*

Proof. Take an instance I_2 which contains three requests $q_0 := (0, 1, 1, \epsilon)$, $q_1 := (0, 2, 1, 1)$ and $q_2 := (1, 2, 1, 1)$ (Figure 11).

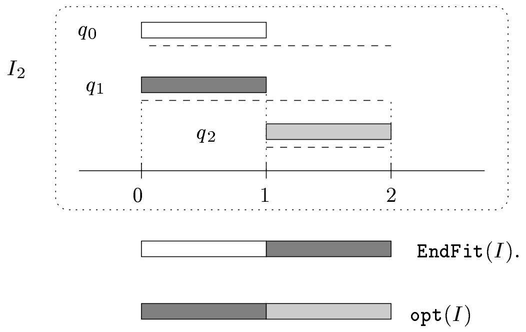


Figure 11: This example demonstrates that the competitive ratio of `EndFit` is at least 2.

Q.E.D.

Both examples are at the extreme end of each other in the sense that `Greedy`(I_2) and `OptFit`(I_1), on the contrary, are the off-line optimal schedule. Thus, an extension of `Greedy` and `OptFit` is the randomized scheduler `optXgdy` who randomly switch mode between `Greedy` and `OptFit`.

It is interesting to find the competitive ratio of `Greedy`, `OptFit` and `optXgdy`.

4 System Design and Implementation

Lets specify, from the viewer point of view, the features to be provided by the system.

Features of system. The viewer want to view a large image (for example, a $4000 \times 4000 \times 24$ bits image), which is stored in a server connected by a thinwire. As the image is large, the whole image could not be displayed on the displaying window. Call the image displayed the *viewing image*. Call the boundary of the viewing image with respect to the original image the *viewing boundary*. (Figure 12).

After connection to the server is established, a low resolution image is first sent over and displayed on the display window. Now, the viewer could perform the following operations: moving, zooming, panning and jumping on the image.

Moving	The viewer could move the mouse/cursor pointer on the image.

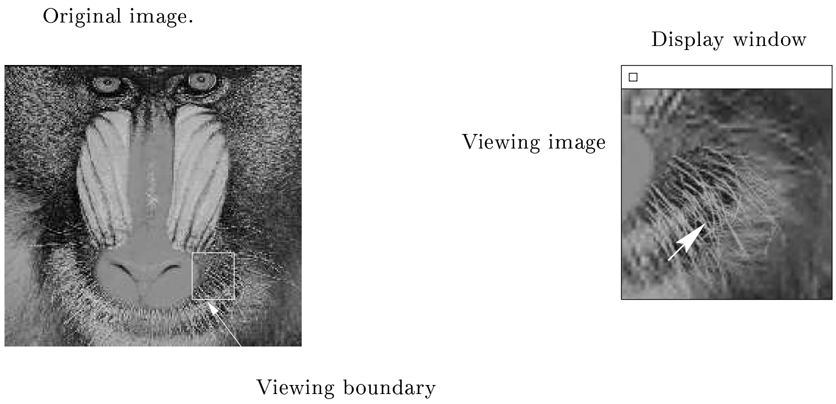


Figure 12: An example of display window, viewing window and viewing image.

Zooming in/out — The viewer could zoom the image. In other words, the viewing boundary could be shrunk or enlarged while keeping the size of the display window unchanged.

Panning — The viewer could smoothly move the viewing boundary to a new location while keeping its size unchanged.

Jumping — The viewer could set the the top-right corner of the viewing boundary to a new location, regardless of its previous location.

Details around the cursor pointer are sent across continuously and they are filled in continuously into the viewing image.

The system should be "responsive" in the sense that the four operations must be carried out immediately.

System Design There are two main concerns in the system design.

1. Though we have not decided on the representation of a multi-foveated image, it is clear that the representation will be sparse. The question here is how to design an efficient sparse data-structure that support fast rendering of foveated images.

2. The main I/O channels include the networks, input device and display unit. We need a way to coordinate the uses of these channels.

References

[1] K.J. Wiebe A. Basu. Videoconferencing using spatially varying sensing with multiple and moving fovea. In *Proceedings of the IEEE International Conference on Pattern Recognition*, volume 3, pages 30–34, 1994.

[2] C.H. Anderson, P.J. Burt, and G.S. van der Wal. Change detection and tracking using pyramid transform techniques. *Proceedings of the DARPA Image Understanding Workshop*, pages 72–78, 1985.

[3] A. Basu, A. Sullivan, and K.J. Wiebe. Variable resolution teleconferencing. In *Proceedings of the IEEE International Conference on Pattern Recognition*, volume 3, pages 170–175, 1994.

[4] B. Bederson, R.S. Wallace, and E.L. Schwartz. A miniaturized active vision system. In *11th IAPR International Conference on Pattern Recognition*, pages 58–62, The Hague, Netherlands, 1992. Specialty Conference on Pattern Recognition Hardware Architecture.

[5] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms i. *Communications in Pure and Applied Mathematics*, XLIV:141–183, 1991.

[6] G. Beylkin, R. Coifman, and V. Rokhlin. Wavelets in numerical analysis. In *Wavelets and Their Applications*, pages 181–210. Jones and Bartlett, 1992.

[7] P.J. Burt. Smart sensing within a pyramid machine. *Proceedings of the IEEE*, 76(8):1006–1015, 1988.

[8] P.J. Burt and Theodore Adelson. A laplacian pyramid for data compression. *IEEE Trans. Commun.*, C-8:1230–1245, 1981.

[9] Ingrid Daubechies. *Ten Lectures on Wavelets*. SIAM, 1992.

[10] D.H.Hubel and T.N.Wiesel. Receptive fields and functional architecture of monkey striate cortex. *J. Physiol. (Lond.)*, 195:215–243, 1968.

[11] A. Eleftheriadis and A. Jacquin. Automatic face location detection and tracking for model-assisted coding of video teleconferencing sequences at low bitrates. *Signal Processing: Image Communication*, 7(3):231–248, 1995.

[12] R.A. Fisher and H.M. Tong. A full-field-of-view dome visual display for tactical combat training. In *Proc. Image Conference IV*, Phoenix, Arizona, June, 1987.

[13] F.Panerai, C.Capurro, and G.Sandini. Space variant vision for an active camera mount. In *Proc. SPIE AeroSense95*, 1995.

[14] M. Frazier and B. Jawerth. Applications of the ϕ and wavelet transforms to the theory of function spaces. In *Wavelets and Their Applications*, pages 377–418. Jones and Bartlett, 1992.

[15] B. Girod. Eye movements and coding of video sequences. *SPIE Visual Communications and Image Processing*, 1001:398–405, 1988.

[16] G.G. Holmes. The cortical localization of vision. *Br. Med. J*, ii:193–199, 1919.

[17] Nuggehally S. Jayant and Peter Noll. *Digital coding of waveforms: principles and applications to speech and video.* Englewood Cliffs, New Jersey, 1984.

[18] Philip Kortum and Wilson S. Geisler. Implementation of a foveated image coding system for image bandwidth reduction. In *Human Vision and Electronic Imaging, SPIE Proceedings Vol. 2657*, pages 350–360, 1996.

[19] M. Levoy and R. Whitaker. Gaze-directed volume rendering. *Computer Graphics*, 24(2):217–223, March 1990.

[20] Richard J. Lipton and Andrew Tomkins. Online interval scheduling. *Proceedings of the 5th annual ACM-SIAM Symposium on Discrete Algorithms*, pages 304–311, 1994.

[21] S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattern Anal. Machine Intell.*, 11:674–692, 1989.

[22] S. Mallat, Z. Zhang, and G. Papanicolaou. Adaptive covariance estimation of locally stationary processes. *Annals of Stat.*, 1997 (To appear).

[23] Stephane Mallat. *A Wavelet Tour of Signal Processing.* 1997.

[24] M. Manasse, L.A. McGeoch, and D. Sleator. Competitive algorithms for server problems. In *Proc. 20th Annual ACM Symposium on Theory of Computing*, pages 322–333, 1988.

[25] David Marr. *Vision: a computational Investigation into the Human Representation and Processing of Visual Information.* W.H. Freeman, 1982.

[26] J. Max. Quantizing for minimum distortion. *IRE Transactions on Information Theory*, IT-6(1):7–12, 1960.

[27] B.H. McCormick, T.A. DeFanti, and M.D. Brown. Visualization in scientific computing. *ACM Computer Graphics (special issue)*, 21, 1987.

[28] Y. Meyer. *Wavelets and Operators*. London Math. Society, 1989.

[29] M. Minkowski. Experimentelle untersuchungen uber die beziehungen der grosshirninde. *Arb. Hirnanat. Inst. Zurich*, 7:259, 1913.

[30] Alireza Moini. Vision chips or seeing silicon. www page at url:⟨http://www.eleceng.adelaide.edu.au/ groups/gaas/bugeye/visionchips/⟩, Last update April 1997.

[31] Alireza Moini. Vision chips or seeing silicon. Technical Report, Center for High Performance Integrated Technologies and Systems, The University of Adelaide, March 1997.

[32] S. Poliak. The main afferent fibre systems of the cerebral cortex in primates. *Univ. Calif. Publ. Anat.*, 2:107–207, 1932.

[33] E.L. Schwartz. Spatial mapping in primate sensory projection: analytic structure and relevance to perception. *Biological Cybernetics*, 25:181–194, 1977.

[34] Eric L. Schwartz. Topographic mapping in primate visual cortex: History, anatomy, and computation. In D. H. Kelly, editor, *Visual Science and Engineering: Models and Applications*, pages 293–360. Marcel Dekker, 1994.

[35] C.E. Shannon. Communications in the presence of noise. In *Proceedings of the IRE*, volume 37, pages 10–21, January 1949.

[36] J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445–3462, December 1993.

[37] D. Sleator and R. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208, 1985.

[38] K.R. Sloan and S.L. Tanimoto. Progressive refinement of raster images. *IEEE Trans. Comput.*, C-28(11):871–874, 1979.

[39] Wavelet technology and information access. www page at url:⟨http://www.summus.com/⟩, Summus, Ltd, Last update November 21,1997.

[40] S.A. Talbot and W.H. Marshall. Physiological studies on neural mechanisms of visual localization and discrimination. *Am. J. Ophthalmol*, 24:1255–1263, 1941.

[41] T.H.Reeves and J.A.Robinson. Adaptive foveation of mpeg video. *Proceedings, 4th ACM International Multimedia Conference*, 1996.

[42] H.M. Tong and R.A. Fisher. Progress report on an eye-slaved area-of-interest visual display. In *Proc. Image Conference III*, Phoenix, Arizona, May, 1984.

[43] K.H. Tzou. Embedded max quantization. In *Proc. IEEE Int. Conf. Acoust. Speech. Sig. Proc.*, pages 505–508. ICASSP'86, 1986.

[44] Kou-Hu Tzou. Progressive image transmission: a review and comparison of techniques. *Optical Engineering*, 26(7):581–589, 1987.

[45] G.K. Wallace. The jpeg still picture compression standard. *Communications of the ACM*, 34(4):30–44, April 1991.

[46] R.S. Wallace, P.-W. Ong, B.B. Bederson, and E.L. Schwartz. Space variant image processing. *Intl. J. of Computer Vision*, 13(1):71–90, 1994.

[47] D. Whitteridge and P.M. Daniel. The representation of the visual field on the cerebral cortex in monkeys. *J. Physiol. London*, 159:203–221, 1961.

[48] R.C. Wood. On optimum quantization. *IEEE Transactions on Information Theory*, IT-15(2):248–252, 1969.

[49] Y. Yeshurun and E.L. Schwartz. Shape description with a space-variant sensor: Algorithms for scan-path, fusion and convergence over multiple scans. *IEEE Trans. Pattern Anal. Machine Intell.*, PAPMI-11:1217–1222, 1989.