# A Framework for Optimization of a Multiresolution Remote Image Retrieval System

A. Ortega, Z. Zhang

Dept. of Electrical Eng. and CTR
Columbia University
New York, NY 10027

M. Vetterli

Dept. of EECS
University of California
Berkeley, CA 94720

## Abstract

*In this paper, we study the tradeoffs involved in choosing the bit allocation in a multiresolution remote image retrieval system. Such a system uses a multiresolution image coding scheme so that a user accessing the database will first see a coarse version of the images and will be able to accept or discard a given image faster, without needing to receive all the image data. We formalize the problem of choosing the bit allocation (e.g., in the two resolution case, how many bits should be given to the coarse image and the additional information, respectively?) so that the overall delay in the query is minimized. We provide analytical methods to find the optimal solution under different configurations and show how a good choice of the bit allocation results in a significant reduction of the overall delay in the query (by up to a factor of two in some cases).*

## 1 Introduction

Consider a generic multiresolution (MR) remote image retrieval system (see [1] for an example of such a system). The multiresolution approach is already being used for commercial products (e.g. Kodak's Photo CD) and has also been proposed for retrieval of video [2]. Users accessing the system will be searching for one or more images within those available in the remote database. The two main components of the system are an image database and a user interface which handles the communication resources transparently to the user. We assume that there are two main stages in a query: (i) the *database search* stage, where in response to the user specification the database manager defines a set of possible candidate images, and (ii) the *browsing* stage, where the user tries to select one or more candidate images, called *target images.* In the latter stage the user is presented with a set of low resolution images (e.g. icons), and can then view them at increasing resolutions, up to the highest available quality, and this until one or more images are selected or the query is terminated. The motivation is that by having fast access first to "coarse" versions of the images, users are allowed to discard, if desired, some of the images *without necessarily having to receive the full quality image*, thus reducing the overall transmission costs of the system. When favoring an MR approach, the underlying assumption is that *the communication costs are the limiting factor.* This situation arises either because (i) the users have access to low-speed (or shared) links, so that transmission delay dominates the total delay in the query (over, for instance, the delay introduced by the search within the database) or simply because (ii) the system has to be designed to minimize the total transmission cost, which we assume to be proportional to the transmission time.

In this work we will concentrate on the browsing stage of the queries. We will further assume that browsing and database search are independent so that our optimization of the browsing stage will not affect the performance of the database search stage. While work reported in the literature has focused on the progressive image transmission schemes [3, 4] here we look at the image coding scheme from a systems perspective. Images in the database are coded with an MR scheme (which we do not specify) so that, taking the two-resolution case as an example, at the start of the browsing stage a fraction $\alpha B$, $0 < \alpha < 1$, of the $B$ bits of the image is transmitted and a low resolution image is reconstructed using those bits. The remaining $(1-\alpha)B$ needed to reconstruct the full resolution image will only be sent if the user requests it. We tackle the problem of assigning a number of bits to each of the image layers (i.e. in our example choosing $\alpha$) so that the performance of the image retrieval system is optimized.

Note that in a typical bit allocation problem for an MR image coder [5] the objective is to assign bits to each of the image layers to maximize the full quality and possibly to meet some intermediate quality objectives. However here our concern is to study how the bit allocation among the successive image layers affects the overall system performance. As an example, in [1] arbitrary compression rates are chosen for the different resolutions: we point out that this choice can be made so that the system performance is optimized.

To clarify the scope of our optimization, let us note that we can divide the resources used in an MR image retrieval system into roughly three groups: (i) the database computation resources, (ii) the communication resources, and (iii) the computation (including memory) resources at the user sites. We will only consider the latter two resources, under the assumption that the bit allocation only affects the browsing stage and not the search within the database. We can thus state the problem we are seeking to solve as follows, imposing the constraint that all the images in the database use the same allocation:

**Problem 1** *How do we allocate the bits to each of the image layers to minimize the total transmission delay or, equivalently, the transmission cost, during a query for a target image.*

At the beginning of the browsing stage, the user is provided with a set of icons from which to select the target image. Since the icons will have very low resolution, it will typically be hard to determine whether the icon set contains a target image and thus the user will have to retrieve some of the images at increasing resolutions in order to make a choice. The trade-off that arises in choosing the bit allocation is clear. If the intermediate resolution were of very high quality (thus requiring a large number of bits, or $\alpha$ close to 1), the user would be able to make a decision on whether the image is acceptable but the cost of retrieving non-acceptable images would be high. Conversely, if the intermediate quality were low (and thus the required bit rate were small, or $\alpha$ close to 0) a decision on the image would not be easy while the cost for choosing a "wrong" icon would be small. The aim of this work is to analyze the trade-off.

This paper is organized as follows. Section 2 provides a more detailed description of a multiresolution image retrieval system and formally defines the parameters of the system as well as our objective function. Section 3 provides solutions to the problem under different configurations. Section 4 draws conclusions and points out areas for further work.

## 2 System definition

### 2.1 Multiresolution browsing

Consider the following flow diagram for the user interaction (see Fig. 1). Each user first generates a request and, after a database search, a set of low resolution candidate icons is displayed at the terminal. The user then, at *Stage 1*, selects one of the icons so that its corresponding low resolution image is displayed on the terminal. At *Stage 2*, if (a) the quality of the low resolution image is too poor to decide or (b) the image seems to be adequate for the user requirement, the user requests that the additional information (necessary to create the full resolution picture) is sent (go to *Stage 3*). Otherwise, if the displayed image has sufficient quality and is not one of the targets, it is rejected and another icon is selected (go back to *Stage 1*). At *Stage 3* the full resolution image is displayed and the user can accept it (and terminate the query) or reject it and select another icon (go back to *Stage 1*). The process repeats until an appropriate image is found.

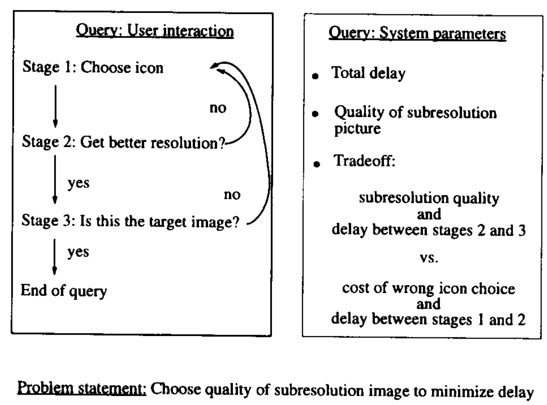


Figure 1: Multiresolution image retrieval system: typical user interaction and corresponding system parameters.

Note that the above description presents a somewhat simplified user interaction since only one candidate image can be considered at any given time. A more general case would not have such a restriction and users would be allowed to store images at different resolutions and then make their decision by comparing those selected. The search can be seen as a process where the user accumulates images at different resolutions (from icon up to full resolution) until the target (one or several images) has been found. While a system with memory might seem more realistic, our results indicate that, as far as the allocation is concerned, the results are identical in both the memory and memoryless cases (see [6] for details).

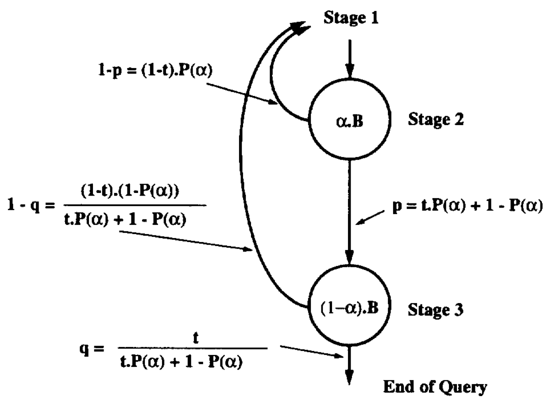


Figure 2: System model for a multiresolution image retrieval system. $t$ is the probability an image is one of the targets. $P(\alpha)$ is the probability that $\alpha$ percent of the total bits provide sufficient quality. $B$ is the image size.

## 2.2 System Model

The previous system description can be formalized as follows (refer to Fig. 2). Let $t$ be the probability that an image chosen from the set of icons is one of the *target* images. Let $\alpha$ denote the percentage of the image data volume in the low resolution; we assume that all images are coded using the same parameter $\alpha$. Let $P(\alpha)$ denote the probability that the quality of the image reconstructed using $\alpha$ percent of the bits is sufficient to make a correct decision (see Section 2.3). Our objective is to obtain $\alpha_{opt}$, the optimal value of $\alpha$ such that the mean response time is minimized, where the response time is defined as the time interval from the time the request is generated until the time the target image is found.

We model the user interaction (refer again to Fig. 2) by assigning probabilities to the transitions between the successive stages of the query as follows. A transition from *Stage 2* to *Stage 1* occurs when the image has sufficient quality but is not a target, with probability

$$1 - p = P_{2\to1} = (1-t)\cdot P(\alpha).$$

A transition from *Stage 2* to *Stage 3* occurs if (a) the image has insufficient quality or (b) if a target image has been found, with probability

$$p = P_{2\to3} = 1 - P(\alpha) + t\cdot P(\alpha).$$

Finally at *Stage 3*, the query will end if a target image has been found and will go back to *Stage 1* otherwise, so that we have:

$$1 - q = P_{3\to1} = \frac{(1-t)\cdot(1-P(\alpha))}{t\cdot P(\alpha) + 1 - P(\alpha)}, \quad \text{and}$$

$$q = P_{3\to e} = \frac{t}{t\cdot P(\alpha) + 1 - P(\alpha)}.$$

## 2.3 Probability of sufficient quality

Given a set of $N$ images, $\mathcal{S}$, assume that we allocate to all of them the same $\alpha$. We propose to model $P(\alpha)$, the probability that an image, picked at random from the set, has "sufficient" quality for the user to make a decision, as follows. To each image from the set $s_i \in \mathcal{S}$, we can associate a rate-distortion (R-D) characteristic, where each R-D point corresponds to the image coded at one of the available resolutions. $P(\alpha)$ could be obtained as an average of normalized distortion (for the normalized rate $\alpha$) measured over the image set (for details see [6]).

In the rest of this work we will assume that the probability function $P(\alpha)$ is in the form of $1-(1-\alpha)^m$, where $m$ is a positive integer. Note that our choice is reasonable when considering typical rate-distortion characteristics and it only affects the exact value of our result; the general analysis holds for more general expressions of $P(\alpha)$.

# 3 Analysis and Results

We now provide solutions to the optimization problem outlined in the previous section. Note that we formulated the problem of a single user having access to the database but we now consider the possibility of several users sharing the system. Different methods are called for depending on the exact formulation, in particular whether the communication resources are shared or not. However, we will show that as far as the optimal point is concerned, most cases of interest yield the same solution. Note also that here we assume $P(\alpha)$ to be the same for all users accessing the database. This does not mean that all users are supposed to access the same set of images, rather it implies that all image sets are similar as far as their quality-rate trade-off. We tackle the problem within a dynamic, queueing framework where our objective is to minimize the expected value of the delay for a given user. An alternative solution, based on a "static" average analysis, is also possible and yields the same results (details can be found in [6]).

### 3.1 Separate channels and large image set

We assume here that the $M$ users in the system do not share the communications resources and have only to share the computation resources of the database. The average size of an image is assumed to be $1/\mu_2$. We assume that on average a delay of $1/\mu_1$ is incurred every time a user goes back to Stage 1 in the query; the cost reflects the computation needed in the database to have the next selected image ready for tramission (e.g. encoding, loading into appropriate buffers, etc). Note that although transmission of the icons themselves would produce some delay we are assuming that the icon size is constant and thus we are not including this factor in the optimization process. We assume that the image sets that are being searched are large enough that $t$ is constant during the search. For a given value of $\alpha$, we are interested in the average response time for a user to search for the target image. We model the system as a closed queueing network with four queues, depicted in Fig. 3. The first queue "stores" the users that are currently idle and waiting to generate another query. The remaining queues correspond, respectively, to Stages 1 to 3 in our model. For simplicity, we assume that both image size and the computation delay are exponentially distributed with respective means $1/\mu_2$ and $1/\mu_1$.

Since all the $i_1$ users at stage 1 share the computation resources, the processing rate for one user is $\mu_1/i_1$. The processing rate at which at least one user finishes the process is $i_1(\mu_1/i_1) = \mu_1$. Users at stages 2 and 3 have their own communication channels, with transmission rates $\mu_2/\alpha$ and $\mu_2/(1-\alpha)$, respectively. If there are $i_2$ users at stage 2 ($i_3$ users at stage 3, respectively), the rate at which at least one user finishes transmitting is $i_2\mu_2/\alpha$ at stage 2 ($i_3\mu_2/(1-\alpha)$ at stage 3, respectively).

In the following, we derive an expression for the mean request delay using the Norton equivalent theorem of queueing networks [7]. The Norton equivalent network with a total of $M$ users is shown in Fig. 4.

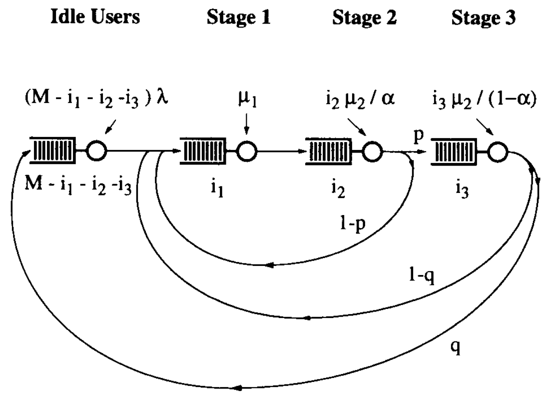


Figure 3: Model for the system as closed queueing network.

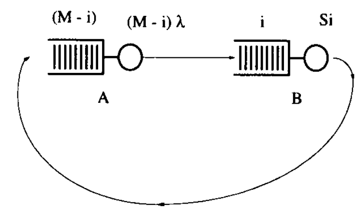


Figure 4: Norton equivalent network.

To find the state-dependent service rate, $s_i$, we use the approach presented in [7]. Let

$$\mu_1(i) = \mu_1, \quad \mu_2(i) = i\mu_2/\alpha, \quad \mu_3(i) = i\mu_2/(1-\alpha)$$

$$X_i(k) = \prod_{j=1}^{k} \frac{y_i}{\mu_i(j)}, \quad i = 1,2,3, \quad k = 0,1,...,M,$$

where $y_1 = y_2 = 1/pq, y_3 = 1/q, y_4 = 1$ is a solution of the balance equation (equation (4) in [7]) and $p$, $q$ are as defined in Section 2.2.

Let

$$G_1(k) = X_1(k)$$

$$G_2(k) = \sum_{i=0}^{k} G_1(i)X_2(k-i), \; k = 0,1,....,M, \quad (1)$$

$$G_3(k) = \sum_{i=0}^{k} G_2(i)X_3(k-i)$$

The state-dependent service rate, $s_i$, is given by

$$s_i = \frac{G_3(i-1)}{G_3(i)}.$$

If we define the state of the system as the number of requests at buffer $B$ in Fig. 4, the state process is a finite population birth-death process with birth-death rates given by

$$\lambda_i = \begin{cases} (M-i)\lambda & 0 \le i \le M-1 \\ 0 & i \ge M \end{cases}$$

and

$$s_i = \frac{G_3(i-1)}{G_3(i)}, \quad 1 \le i \le M,$$

respectively, where $G_3(i)$ is given in (1).

The steady-state mean queue size and request delay can easily be obtained and are given by [8]:

$$E[q] = P_0 \sum_{i=1}^{M} i \prod_{j=0}^{i-1} \frac{\lambda_j}{s_{j+1}} \tag{2}$$

and

$$E[d] = \frac{E[q]}{\lambda(M - E[q])} \tag{3}$$

respectively, where

$$P_0 = (1 + \sum_{i=1}^{M} \prod_{j=0}^{i-1} \frac{\lambda_j}{s_{j+1}})^{-1}.$$

The optimal value of $\alpha$ is found by minimizing $E[d]$ over $\alpha$, $0 \le \alpha \le 1$. The results are summarized in Figs. 5, 6, 7. The most important point is to note that the optimal operating point is not a function of $t$, the number of users $M$ or $\mu_2$. Fig. 5 shows the delay vs. $\alpha$ tradeoff for two values of $t$. The relative gain of using the $\alpha_{opt}$ is nearly the same in both cases. Fig 6 shows the same tradeoff for different values of $\mu_2$. Note that in the bottom two curves $\mu_2 \ll \mu_1$ and therefore the delay due to the image transmission dominates the delay due to the database access. However, for $\mu_2 = 0.01$ the dominant term is the database delay and little can be gained by choosing a correct $\alpha$. As was to be expected, optimizing $\alpha$ only makes sense when communication resources are the bottleneck. In Fig 7 the service rate for the transmission is only ten times slower than that of the database access and we can see that when the number of users increases over ten the dominating factor becomes the database access delay, and therefore the choice of $\alpha$ does not make as much of a difference (because the users share the database access but not the communication resources).

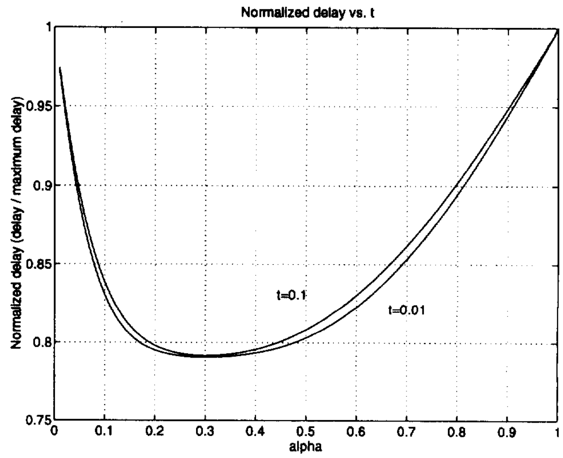


Figure 5: Total delay as a function of $\alpha$ for two values of $t$. In all cases we have that $\alpha_{opt} = 0.3012$. The other parameters are set to $M = 10, m = 5, \mu_1 = 0.1, \mu_2 = 0.01, \lambda = 0.1$. Note that the trade-off is practically identical for both values of t.

### 3.2 Separate channels and small image set: non constant t

The results in the previous section indicate that the value of the optimal $\alpha$ does not change with the number of users in the system. In this section, we consider only one user. There are initially $N_0$ unsearched icons but now we assume that $N_0$ is "small", so that the probability that one chooses the right icon among $i$ unsearched icons is assumed to be $t(i)$, a function of $i$. In the following, we will derive an expression for the average delay, $E_\alpha(i)$, incurred in searching the target image, given there are $i$ unsearched icons. Using renewal theory, we have

$$\begin{aligned} E_\alpha(i) &= 1/\mu_1 + \alpha/\mu_2 + (1-t(i))P(\alpha)E_\alpha(i-1) \\ &\quad +(t(i)P(\alpha) + (1-P(\alpha))) \\ &\quad ((1-\alpha)/\mu_2 + (1-t(i)/(t(i)P(\alpha)+(1-P(\alpha))))E_\alpha(i-1)) \\ &= (1-t(i))E_\alpha(i-1) + \frac{1}{\mu_1} + \frac{1}{\mu_2} - \frac{1}{\mu_2}(1-t(i))P(\alpha)(1-\alpha) \end{aligned} \tag{4}$$

and

$$E_\alpha(1) = \frac{1}{\mu_1} + \frac{1}{\mu_2}. \tag{5}$$

An explicit expression of $E_\alpha(i)$ can be obtained iteratively from (4) and (5) and is given by

$$E_\alpha(i) = (\frac{1}{\mu_1} + \frac{1}{\mu_2})$$

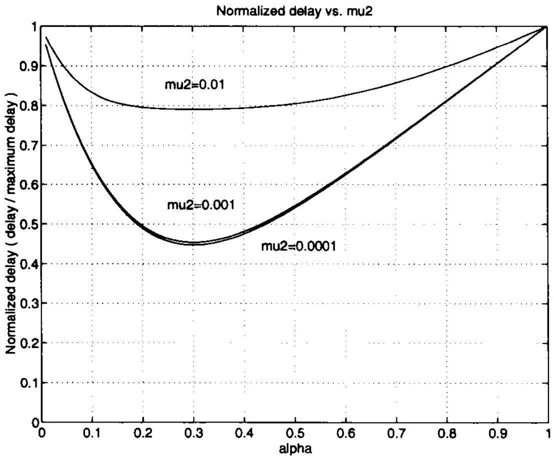


Figure 6: Total delay as a function of $\alpha$ for several values of $\mu_2$. In all cases we have that $\alpha_{opt} = 0.3012$. The other parameters are set to $M = 10, m = 5, \mu_1 = 0.1, t = 0.05, \lambda = 0.1$. Note that for $\mu_2 = 0.01$ the delay due to the database access, $\mu_1 = 0.1$ is still significant so that optimizing the transmission results in modest gains. Conversely for the other two values of $\mu_2$ transmission dominates the delay.

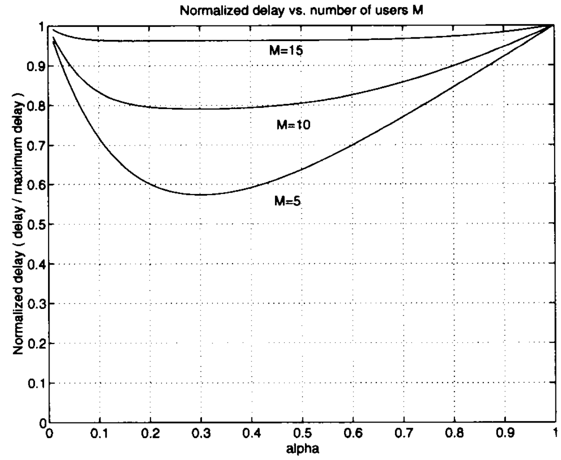


Figure 7: Total delay as a function of $\alpha$ for several values of the number of users. In all cases we have that $\alpha_{opt} = 0.3012$. The other parameters are set to $m = 5, \mu_1 = 0.1, \mu_2 = 0.01, t = 0.05, \lambda = 0.1$. Note that, as all users share the database, increases in $M$ imply that the database delay becomes more significant and the gains obtained by optimizing the transmission are smaller.

$$(1+\sum_{j=2}^{i}\prod_{k=j}^{i}(1-t(k))) - \frac{1}{\mu_2}P(\alpha)(1-\alpha)\sum_{j=2}^{i}\prod_{k=j}^{i}(1-t(k)) \tag{6}$$

Since $t(i) \leq 1$ for all $i$, $1 \leq i \leq N_0$, we have $\sum_{j=2}^{i}\prod_{k=j}^{i}(1 - t(k)) \geq 0$. Therefore, the problem of minimizing $E_\alpha(i)$ subject to $0 \leq \alpha \leq 1$ is equivalent to the problem of maximizing $P(\alpha)(1-\alpha)$ subject to $0 \leq \alpha \leq 1$.

So that we have

$$\alpha_{opt} = \arg \max_{0 \leq \alpha \leq 1}(P(\alpha)(1 - \alpha)). \tag{7}$$

For the same set of parameters, $\alpha_{opt}$ is exactly the same as that obtained in the previous section (although here our analysis only covers the single-user case).

### 3.3 Shared Resources

We now consider the case where all the users share one communication channel. As in Section 3.1, the system can be modeled as a closed queueing system (see Fig. 3), with modified transmission rates which can be determined as follows. The number of users at stages 2 and 3 in Fig. 3 represents the number of users sharing the transmission link. At any given time, a user can either be in stage 2 or stage 3, but cannot be in both at the same time. Therefore, since the link is shared, if there are $i$ and $j$ users at stages 2 and 3 respectively, the transmission rates for one user would be $\frac{\mu_2}{(i+j)\alpha}$ and $\frac{\mu_2}{(i+j)(1-\alpha)}$ at stages 2 and 3, respectively. The rate at which at least one user finishes transmitting would be $\frac{i\mu_2}{(i+j)\alpha}$ and $\frac{j\mu_2}{(i+j)(1-\alpha)}$ at stages 2 and 3, respectively.

Because the transmission rate depends on the number of users at other queueing systems, we cannot use the Norton equivalent theorem of queueing networks. Instead, we solve the steady state probability directly. We define the system state as $(x_1, x_2, x_3)$, where $x_1, x_2$, and $x_3$ denote the numbers of users at the last three queueing systems in Fig. 3, respectively, and $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$, $x_1 + x_2 + x_3 \leq M$. The one step transition probability, $p(i_1, i_2, i_3|j_1, j_2, j_3) = p(x_1 = i_1, x_2 = i_2, x_3 = i_3|x_1 = j_1, x_2 = j_2, x_3 = j_3)$, $0 \leq i_1 + i_2 + i_3 \leq M$, $0 \leq j_1 + j_2 + j_3 \leq M$, is given as follows:

$$p(i_1 + 1, i_2, i_3|i_1, i_2, i_3) = (M - i_1 - i_2 - i_3)\lambda$$

$$p(i_1 - 1, i_2 + 1, i_3|i_1, i_2, i_3) = \mu_1$$

$$p(i_1, i_2 - 1, i_3 + 1|i_1, i_2, i_3) = p\frac{i_2\alpha}{i_2 + i_3}\mu_2$$

$$p(i_1+1, i_2-1, i_3|i_1, i_2, i_3) = (1-p)\frac{i_2\alpha}{i_2+i_3}\mu_2$$

$$p(i_1+1, i_2, i_3-1|i_1, i_2, i_3) = (1-q)\frac{i_3(1-\alpha)}{i_2+i_3}\mu_2$$

$$p(i_1, i_2, i_3-1|i_1, i_2, i_3) = q\frac{i_3(1-\alpha)}{i_2+i_3}\mu_2$$

$$p(i_1, i_2, i_3|i_1, i_2, i_3) =$$

$$1-(M-i_1-i_2-i_3)\lambda-\frac{i_2\alpha}{i_2+i_3}\mu_2-\frac{i_3(1-\alpha)}{i_2+i_3}\mu_2-I(i_1)\mu_1$$

$$p(j_1, j_2, j_3|i_1, i_2, i_3) = 0 \quad \text{for other values of } j_1, j_2, j_3$$

where $I(x)$ is the indicator function, i.e., $I(x) = 1$ if $x > 0$ and $I(x) = 0$ if $x = 0$. The steady state probability $\pi(i_1, i_2, i_3) = p(x_1 = i_1, x_2 = i_2, x_3 = i_3)$ can be obtained by solving the balance equations and the normalized equation, $\sum \Pi(i_1, i_2, i_3) = 1$. The total number of states is $M(M^2+6M+11)/6$. The mean delay is given by

$$E[d] = \frac{E[q]}{\lambda(M - E[q])} \tag{8}$$

where

$$E[q] = \sum_{l=1}^{M} \sum_{i_1+i_2+i_3=l} l\ \pi(i_1, i_2, i_3).$$

Numerical results for $M \leq 10$ indicate that the optimal value of $\alpha$ is independent of the value of $M$ and once again identical to that obtained in Sections 3.1, 3.2. Fig. 8 shows the delay vs. $\alpha$ tradeoff for different number of users, while Fig. 9 shows the tradeoff when $t$ varies.

### 3.4 Discussion

A first conclusion of the foregoing sections is that finding the optimal operating point can be worthwhile in reducing the overall delay, in particular in cases where users are connected to the database through low-speed links. For instance choosing the optimal $\alpha$ can provide reductions in delay of up to a factor of two in the $m = 5$ case (see Figs. 8-9 for the shared resources case). Moreover, the advantage of choosing a correct value for $\alpha$ increases as the parameter $m$, which determines the shape of $P(\alpha)$, increases (details can be found in [6]). In most cases of interest one can expect a relatively large $m$ to be likely, i.e. a relatively small percentage of the total bit rate provides sufficient quality to make a decision.

A second point is to note that the optimal $\alpha$ is independent of the exact procedure that is used for

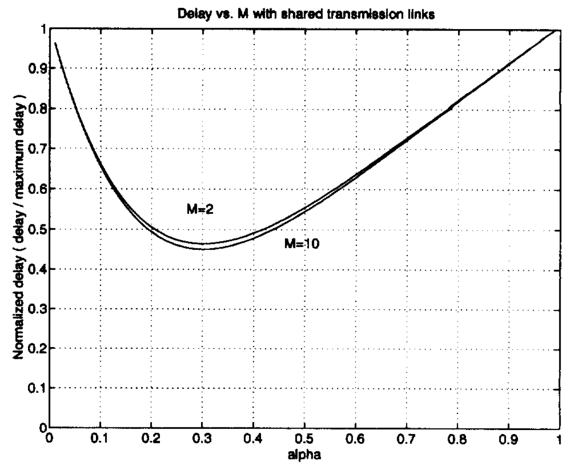


Figure 8: Total delay as a function of $\alpha$ for two values of the number of users when the commnunication resources are shared. In all cases we have that $\alpha_{opt} = 0.3012$. The other parameters are set to $m = 5, \mu_1 = 0.1, \mu_2 = 0.01, t = 0.05, \lambda = 0.1$. The relative gain is practically the same regardless of the number of users because the transmission delay dominates as the links are shared.

transmission. For instance, we find the same results for $\alpha$ whether one or several users access the database, and whether or not the users share the transmission resources. Finally, we see no dependence of the optimal result on the size of the initial image set, or the probability of getting the correct image, $t$.

The intuitive justification is that the exact procedure for retrieving the images is not relevant because we are concerned with minimizing an average cost. Since for every image we have an average measure of the "risk" of having to retrieve the rest of the image (i.e. $P(\alpha)$) and we assume all images are identical (i.e. same probability) it is normal to expect that the only factor to determine the optimal operating point would be $P(\alpha)$.

Similarly, as we increase the number of users, and even if the transmission resources are shared, the optimal value for $\alpha$ remains unchanged. This is again due to our choosing to minimize the average delay for a set of users that are identical, at least in a statistical sense. A "minmax" approach, where the maximum delay instead of the average has to be minimized, would probably yield different results.

Even though the optimal operating point is independent of the system parameters, the gain of using a multiresolution approach is not. In particular

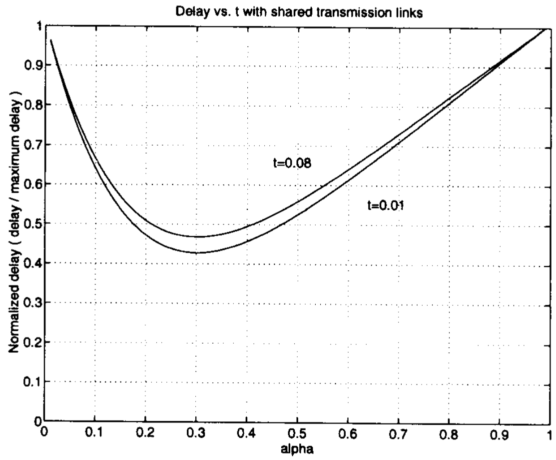


Figure 9: Total delay as a function of $\alpha$ for two values of $t$ when the communication resources are shared. In all cases we have that $\alpha_{opt} = 0.3012$. The other parameters are set to $m = 5, \mu_1 = 0.1, \mu_2 = 0.01, M = 5, \lambda = 0.1$

we pointed how more gain can be expected when the transmission resources are shared or the transmission resources, rather than the database, represent the bottleneck of the system.

## 4 Conclusions and Future Work

In this work, we addressed a problem that arises when designing a remote image retrieval system, namely, that of assigning bits to the different layers of the images to be transmitted. We have solved this problem under assumptions for the average quality of the images ($P(\alpha)$) and the restriction that all the images use the same bit allocation. Results show that significant gain can be expected from choosing a correct bit allocation quite independently of the exact procedure that is used to retrieve the images.

Our analysis leaves a number of questions for future work. In particular it would be of interest to perform quality measures on real images to obtain empirical expressions for $P(\alpha)$. Also, since the average analysis provides the same results as the dynamic one [6], it would be interesting to relax the constraint on the bit allocation and allow each image to have a different $\alpha$. Thus each image would have its own probability of having sufficient quality and thus we could setup a "static" bit allocation problem among the images: to give more bits to those images where the bits can do more "good", in the sense of reducing the overall delay.

## References

[1] N. D. Degan, R. Lancini, P. Migliorati, and S. Pozzi, "Still images retrieval from a remote database: the system imagine," *Signal Processing: Image Communications*, vol. 5, pp. 219–234, May 1993.

[2] T.-C. Chiueh and R. H. Katz, "Multi-resolution video representation for parellel disk arrays," in *Proc. of ACM Multimedia Conf. '93*, (Anaheim, CA), June 1993.

[3] K. Knowlton, "Progressive transmission of grey-scale and binary pictures by simple efficient and lossless encoding schemes," *Proceedings of the IEEE*, vol. 68, pp. 885–896, July 1980.

[4] M. Malak and J. Baker, "An image database for low bandwidth communication links," in *Proc. of the Data Compression Conference*, (Snowbird, Utah), March 1991.

[5] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Trans. on Image Proc.*, 1994. To appear.

[6] A. Ortega, Z. Zhang, and M. Vetterli, "Modeling and optimization of a multiresolution image remote retrieval system," Tech. Rep. CU/CTR/TR 354-94-01, Center for Telecomm. Research, Columbia University, Jan. 1994.

[7] K. Chandy, U. Herzog, and L. Woo, "Parametric analysis of queueing networks," *IBM J. Res. Develop.*, pp. 36–42, January 1975.

[8] D. Gross and C. Harris, *Fundamentals of Queueing Theory*. New York: John Wiley, 2nd ed., 1985.