Multiresolution Cache Management for Distributed Satellite Image Databases Using NACSIS-Thai International Link

Asanobu KITAMOTO
Research and Development Department
National Center for Science Information Systems
2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
kitamoto@rd.nacsis.ac.jp

Abstract

This paper deals with the network technology that will be useful for the development of distributed satellite image databases. In particular, the author proposes a new caching mechanism by combining multiresolution image representation into cache management mechanism. The author first analyzes access pattern to a satellite image database and demonstrates that many kinds of access skew can be well characterized by Zipf-like laws. These statistical properties will be integrated into the cache server specially designed for satellite images. The author then deals with multiresolution image representation, which enables flexible access to huge satellite data; a client can access to any part of the image with many resolution. Finally the author addresses some of the preliminary results on wavelet-based image representation based on multiresolution representation.

1 Introduction

The effective usage of satellite imagery requires solid infrastructure where the retrieval and acquisition of required satellite imagery can be done without requiring much effort to both professional and non-professional users. To realize such useful infrastructure, we should advance research on network technology for the acquisition of large amount of imagery through network.

In this paper, we focus on the development of such network technology, especially focusing on satellite data. By considering the characteristics of satellite imagery, we will first analyze access skew to a satellite image database, and find out that access skew to a satellite imagery can be well characterized by Zipf-like laws. These statistical properties will be integrated in the future development of a cache server specially designed for satellite data.

In addition to access skew addressed above, we are thinking of introducing another type of access skew to a cache server, namely multiresolution cache management. In general, low resolution data has high demand for browsing purpose, so the decomposition of image with respect to resolution causes another type of access skew, which a cache server can take advantage of. Moreover, multiresolution image representation is also required from a user's point of view. Because huge amount of satellite data cannot be downloaded for general users because of relatively limited bandwidth. Also it becomes popular nowadays that users like to view images with progressive transmission. In this way, a user can decide the necessity of the image without receiving all of the image.

Based on the above situation, we investigate technology related to above mentioned research themes. Namely the statistical analysis of access pattern and wavelet-based multiresolution analysis of satellite imagery. Section 2 is the introduction of background on distributed satellite image databases. Section 3 deals with the cache management schemes and analysis of access pattern to satellite images. Section 4, in turn, addresses wavelet transform based multiresolution image representation. Section 5 shows some of the preliminary experimental results, and Section 6 concludes the paper.

2 Distributed Satellite Image Databases

2.1 Background

Satellite image databases are in general distributed over the network. This is because receiving high quality satellite data requires large and expensive special equipment. Hence the number of receiving stations tends to be small due to the limited cost and space. Moreover, the huge size of satellite data prevents long-term archive of satellite data for those without special

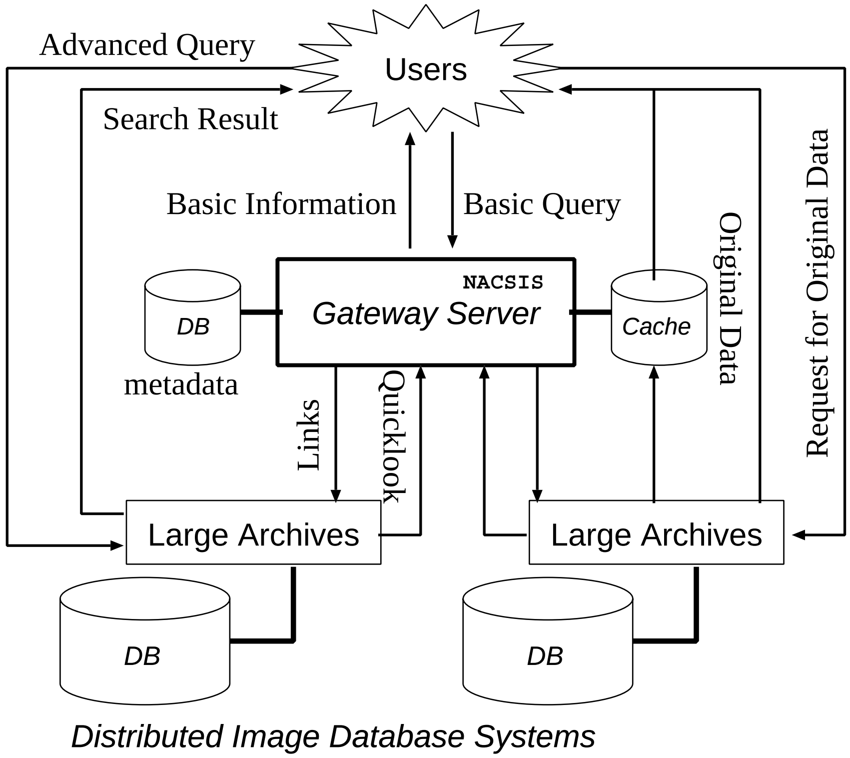


Figure 1: Distributed image databases over the network.

resource in storage and computing power. Even if we could have special computing center that can archive huge collection of satellite data, limited bandwidth of network would prevent intensive exchange of satellite data.

Therefore, the construction of distributed satellite databases is a practical solution for the management of satellite data. Schematic diagram of this configuration is illustrated in Figure 1. In this solution, several receiving stations are located somewhere in the network, providing basic functionality for searching and downloading archived data. However, for a user's point of view, it is more convenient to have a search engine where distributed satellite data can be searched at one place. This front-end server hereafter is called a "gateway server."

This gateway server can also work as a caching server. The combination of a search engine and a caching server works cooperatively as follows. When the gateway server receives requests from clients, the server either return data from its local cache or forward requests to appropriate servers. If the gateway server receives responses to forwarded requests, then the gateway server returns requested data to clients and simultaneously stores the same data into its local cache. This server is expected to work especially effective when remote servers are located in foreign countries, because in general network bandwidth between different countries is narrow enough to become a bottleneck.

2.2 NACSIS-Thai International Link

NACSIS-Thai International Link is one of the international links that we have in mind for the application of the system described above. NACSIS-Thai International Link is an international connection between Japan and Thailand whose bandwidth is 2Mbps. This link is used for exchanging academic information between those countries. Because of its high stability and reliability, this link has attracted many researchers for using it as a testbed for many kinds of applications. In particular, this link witnesses rapid growth in the amount of traffic in satellite data exchange through the network. This requires constant and intensive usage of network. Although 2Mbps is not a very narrow link, the bandwidth of this international link is small compared to SINET (Science Information Network).

Then it is a natural idea to set up a caching server at the entrance to the bottleneck, and improves response to frequently accessed data. In the following, we will address some of the problems for this caching server, and discuss the model of cache management schemes.

2.3 Cache Management for Satellite Imagery

Hence the purpose of the paper is to propose new cache management schemes specially designed for distributed satellite image databases. A few characteristics in the following are worth considering.

1. To deal with the huge size of satellite imagery, all-or-nothing-type decision in removal policy may not be the optimal strategy. Instead, the caching server can archive “a part of “ the image for viewing purpose. This idea can be naturally combined with multiresolution image representation.

2. Usually, a caching server has to handle many kinds of data types, such as text and video. However, a caching server discussed in this paper only deals with “image” type, or a few more at most. Hence the handling of various media types is not an important factor in this framework.

3. Another important distinction that should be mentioned is that satellite imagery is not modified or rarely modified if any. Hence we do not have to take care of modification check. This difference reduces considerable amount of workload that a caching server has to do otherwise.

The last two characteristics affect favorably on our cache management scheme because these characteristics serve to reduce the amount of task. On the contrary the first characteristic, what we call “multiresolution cache management,” adds new workload to the cache management scheme as to handling the resolution scale of each image.

3 Cache Management

3.1 Background

Caching servers help improve efficiency and reliability for data distribution over the network by storing frequently accessed data at locations closer to clients. The explosive growth of the World Wide Web stimulated research on caching servers, and many operational systems have been developed; to name a few, Apache, Squid, etc.

Caching servers may be best characterized by removal policies used for cache management. Since the amount of memory that can be spared for caching is always limited, the removal of ”less important” cached data determines the efficiency of the system. Fundamental strategies often used include LRU (Least Recently Used) or LFU (Least Frequently Used) schemes. There are also many variants of them [1], and they are also extended to incorporating other factors that may lead to inefficiency; for example time required for the transmission of data from the original server to the caching server is often taken into account [10, 9].

The cache management algorithm is essentially equivalent to knapsack problem, where $\mathbf{D} = \{D_1, \cdots, D_N\}$ represents the dataset, and the variable $x_i \in \{0, 1\}$ represents that data D_i is cached $x_i = 1$ or not $x_i = 0$. Moreover, the relevance and the size of data D_i is represented by w_i and s_i respectively. Then the cache management scheme can be formulated as the following knapsack problem:

$$\text{maximize} \qquad V = \sum_{i=1}^{N} x_i w_i \tag{1}$$

$$\text{subject to} \qquad \sum_{i=1}^{N} x_i s_i \leq S \tag{2}$$

From the above equation, it is clear that the assignment of relevance w_i to each data is the essential part of the cache management scheme. In this paper, we will define w_i from the statistics derived from access pattern to satellite image databases.

3.2 Weighting

The relevance of each data D_i can be determined from several statistics. Our hypothesis can be described in the following formula:

$$\begin{aligned} w_i &= w(F_i, T_i^a, T_i^o, L_i) && (3) \\ &= w_1(F_i) w_2(T_i^a) w_3(T_i^o) w_4(L_i) && (4) \end{aligned}$$

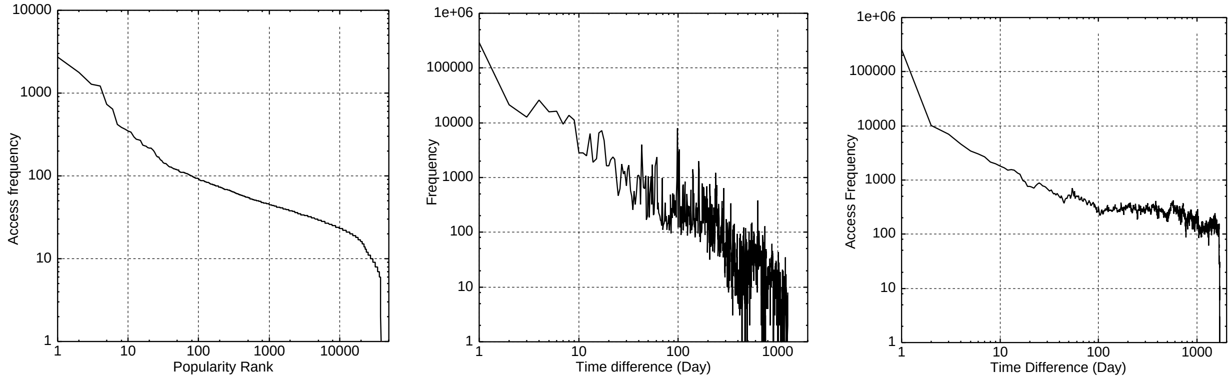
Here Equation (3) denotes that w_i is determined by the probability distribution of four random variables, and in Equation (4) the formula is developed on the assumption of mutual independence of random variables, where the joint distribution of four random variables is represented by the product of each probability distribution. In the following, we will briefly describe four random variables that appear in Equation (3).

1. The first term F_i represents access skew to popular data. Some satellite images contain significant meteorological phenomena, such as typhoons, severe rainfall or snowfall, and this attracts many users for a long duration of time. This variable is the basis of LFU policy.

2. The second term T_i^a represents access skew to more recently *accessed* data. This variable is the basis of LRU policy.

3. The second term T_i^o represents access skew to more recently *observed* data. This sounds similar to the second term, but the meaning is considerably different. Recent satellite data are much more relevant than old ones for most of the users, while only a limited number of users access older data with special interest.

4. The fourth term L_i is related to multiresolution cache management. L_i denotes the resolution scale of the image, which we will introduce in the next section. Roughly speaking, resolution scale



(a) Access Skew to Popular Images

(b) Access Skew to Recently Accessed Images

(c) Access Skew to Recently Observed Images

Figure 2: Various Access Skew Calculated from the Real Data. All figures are shown in log-log scale. Note that in (b) and (c) the label at minimum difference are shown as "1," but it is actually "0," which means the same day. The value on the difference axis should be decreased by one. A fraction is rounded down.

$L_i = 0$ corresponds the most coarse resolution scale and often used for browsing purpose (sometimes called a quicklook image). Incrementing the level by one increases the resolution by a factor of 2. Images of higher level is not transmitted unless a client explicitly request the data.

3.3 Access Skew

In the following, we will review the access patterns, namely the probability distribution of random variables introduced above. The probability distribution is derived from the real access pattern collected at a image database server for satellite images operating at operating at the Institute of Industrial Science, University of Tokyo. This (www / gopher) server keeps its operation since 1995, and access log files are kept since then. Readers are referred to other analysis on these access logs [5].

Many types of access skew is very often referred to by literature on web caching. In characterizing the relative number of requests made to different Web documents, Zipf-like laws are often used to relate frequency P of requests to documents and popularity rank ρ [4, 2]. Here Zipf-like laws are represented by the following formula:

$$P \sim \rho^{-\beta} \tag{5}$$

where the coefficient β is said to be take a value close to one.

3.3.1 Access Skew to Popular Images

Figure 2(a) shows access skew to popular images calculated from 728697 accesses to 38631 images. The graph consists of approximately linear segments on a log-log scale, which clearly suggests Zipf-like laws hold for this access pattern. However, to put it precisely, this graph suggests that actually it is a mixture of multiple Zipf-like laws.

That is, we can divide this graph into three intervals, [1,50], [50, 10000], and [10000, 40000]. It seems that different Zipf-like laws (different coefficient) fit well to each interval. The first interval is well described by Zipf-like function with $\beta = 0.80$, while the second, $\beta = 0.29$. This result suggests that Figure 2 is actually a mixture of two different access patterns. We will discuss this idea later.

3.3.2 Access Skew to Recently Accessed Images

Next we investigate access skew related to recency. Figure 2(b) shows access skew to more recently accessed images calculated from 595789 difference in subsequent access time. The figure illustrates rapid decrease as the time difference between accesses increase. This shows Zipf-like laws with $\beta = 1.35$. In this case, only one Zipf law seems to be enough to describe the access pattern.

3.3.3 Access Skew to Recently Observed Images

Figure 2(c) is calculated from 671013 accesses. This graph again shows a mixture of several Zipf-like laws. The first one is the access to satellite images on the same day, where the number is extraordinary large compared to other time difference. This jump is probably caused by the periodic distribution of satellite images to other sites. The next one is between the interval [1,100], where Zipf-like function with $\beta = 0.98$ well describes the graph. On the contrary, the interval [100, 1500] does not seem to be subject to Zipf-like laws (or Zipf-like laws with $\beta \sim 0$). This is an interesting feature of this graph.

Actually $\beta \sim 0$ corresponds to random access; so the effect of access skew to recency is diminished at about 100 days (a little more than 3 months), and after that no access skew is observed. This change of access pattern may be reason for the mixture of access patterns observed in the access skew to popular images. If we have a lot of random access, less popular data receives more access than expected from the access skew pattern, so we observe an access skew with larger β coefficient, in other words, a heavier tail.

3.3.4 Access Skew to Lower Resolution Images

Low resolution versions of satellite imagery are often sufficient for browsing and quick inspection of image contents. In this case, images for higher resolution are only required for limited region in the whole satellite imagery. This suggests that the efficiency of the caching server shall be improved by exploiting the relationship between image resolution and access frequency. In other words, we can artificially create access skew to low resolution data, and this may lead to more efficient cache management. In this case, the removal policy in multiresolution cache management should favor low resolution imagery to be removed later.

Unfortunately, access skew on this type is not available to us. But if we have to assume some kind of laws to model this access skew without any a priori information, maybe the assumption of another Zipf-like laws may be an appropriate choice considering the ease of computation.

3.3.5 Relevance

In conclusion, four types of access skew pattern can be represented by a combination of Zipf-like laws. However, we also have to consider the effect of random access pattern, whose effect becomes clearer in less popular images. Relevance to each data can be calculated by the product of four terms of Zipf-like laws, each has its own β parameter. Note that we do not need to normalize relevance, because relative magnitude of w_i is important in Equation (1).

4 Multiresolution Image Representation

4.1 Background

Recently we see rapid growth of interest in progressive transmission of images, because this kind of method is efficient for fast inspection of images under network environment. Although progressive transmission requires additional computing power for encoding and decoding, recent growth of CPU power at a client side allows a minor increase in computation, and nowadays network bandwidth may become a relatively scarce source. Because of this situation, progressive transmission becomes an important part of the development of image transmission and of distributed satellite image database systems.

In the progressive resolution transmission scheme, an image with reduced resolution (to be displayed in a small size) is transmitted first. Afterward, the information to obtain images with increasing dimension or resolution (i.e. larger viewing area or pixels/area) from the smaller versions is transmitted [8]. Such progressive image coding schemes have been the subject of research for a long time. Several types of progressive transmission, such as progressive JPEG (Joint Photographic Experts Group) [11] which rearranges the order of DCT coefficients, are already used with wide acceptance. Many other methods are based on pyramid structure, in which low resolution images are generated by averaging (arithmetic or weighted average) a block of pixels (usually 2×2) on a high resolution image. Some of the papers include Yasuda et al. [13], and Burt et al. [3], where in the latter perfect reconstruction of the original image is also possible.

The idea in this paper is to combine progressive transmission methods with caching management schemes for both the reduction of total transmission time and the improvement of cache hit. However, the combination of progressive transmission with caching mechanism is not our innovation. For example, Yasuda et al. [14] proposed a similar algorithm to LRU based on the multiresolution image representation, and cache hit ratio is estimated using Markov model. Moreover, Oguchi et al. [6, 7] investigates quality-based caching for still imagery and video, but specific methods used for quality control is not explicitly discussed, and multiresolution representation is not discussed in a uniform manner.

Hence in this paper, we first test wavelet transform-based multiresolution representation on satellite imagery. Note that in this paper, we are only interested in “lossless” coding of images, or at least coding that allows perfect reconstruction of the original image. This is the requirement often required in satellite images, in which all pixels contain relevant information.

4.2 Wavelet Transform

Recently wavelet transform has stimulated research on progressive transmission, because wavelet transform can be used as a basis of multiresolution representation, and because multiresolution representation has been shown to be naturally suited for progressive transmission. Traditional methods such as Harr bases are one of the special cases of wavelet, and wavelet transform gives us a unified framework for dealing with multiresolution analysis. which has been shown to be naturally suited for progressive transmission. In

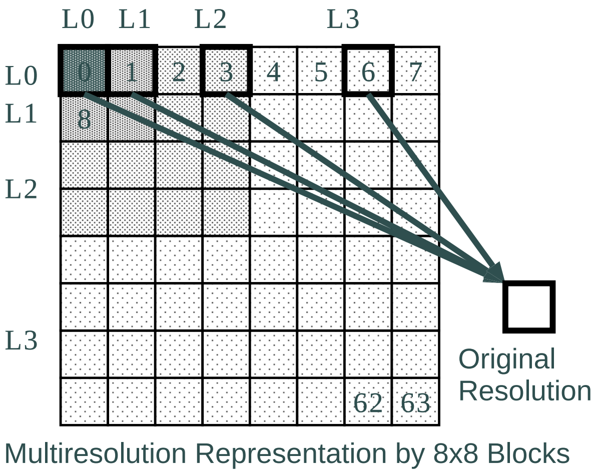


Figure 3: The schematic diagram of the multiresolution image representation.

this paper, we also use wavelet transform for multiresolution image representation, specifically S+P transform proposed by Said et al. [8].

S+P transform is similar to the Haar multiresolution image representation. A sequence of integers $c[n]$, $n = 0, \cdots, N-1$, with N even, can be represented by the two sequences

$$\begin{aligned} l[n] &= \left\lfloor \frac{c[2n] + c[2n+1]}{2} \right\rfloor, \quad n = 0, \cdots, \frac{N}{2} - 1 \\ h[n] &= c[2n] - c[2n+1], \quad n = 0, \cdots, \frac{N}{2} - 1 \end{aligned} \tag{6}$$

where $\lfloor \dot{} \rfloor$ corresponds to downward truncation. The sequences $l[n]$ and $h[n]$ form the S-transform of $c[n]$. The inverse transformation is

$$\begin{aligned} c[2n] &= l[n] + \left\lfloor \frac{h[n] + 1}{2} \right\rfloor \\ c[2n+1] &= c[2n] - h[n]; \end{aligned} \tag{7}$$

The two-dimensional transformation is done by applying the transformation Equation (6) sequentially to the rows and columns of the image. The successive application of this transformation form a hierarchical pyramid. This is of course a perfectly reconstructive coding scheme.

There are several methods for improving coding efficiency of the sequence Equation (6). One basic strategy tested in this paper is to predict $h[n]$ (high frequency component) from several $l[i]$ and $h[i]$ coefficients. In [8] three set of parameters are used for this technique. Then the coefficients are entropy coded using arithmetic coding [12] with MS-VLI representation used.

4.3 Hierarchical Pyramid

Figure 3 shows a hierarchical pyramid and the hierarchical decomposition of spatial frequency. Here the block at upper left corner corresponds to the lowest frequency and shown as resolution scale L_0. This block retains the basic structure of the image and can be used as a quicklook image. On the other hand, the neighboring three blocks, illustrated in hatched squares, correspond to the next lower resolution scale, and represents edges (and others) existing on the smoothed image. Most of the blocks in Figure 3 actually belong to resolution scale L_2. Because of the decimation in the pyramidal structure, the size of the coefficients in higher resolution is always bigger than that in lower resolution. Hence the block of L_2 is divided into 4×4 sub-blocks so that each sub-block has the same size of the block of L_0.

After this division, we put an number to each sub block, the total of 64 sub blocks as shown in Figure 3. Next suppose that we want to see the right upper corner of the image, the following scenario shows how multiresolution progressive transmission works.

1. First a client access this image by requesting a quicklook image, namely Block 0 ($L = 0$). A client looks into the quicklook image, and find something interesting in the upper right corner of the image.

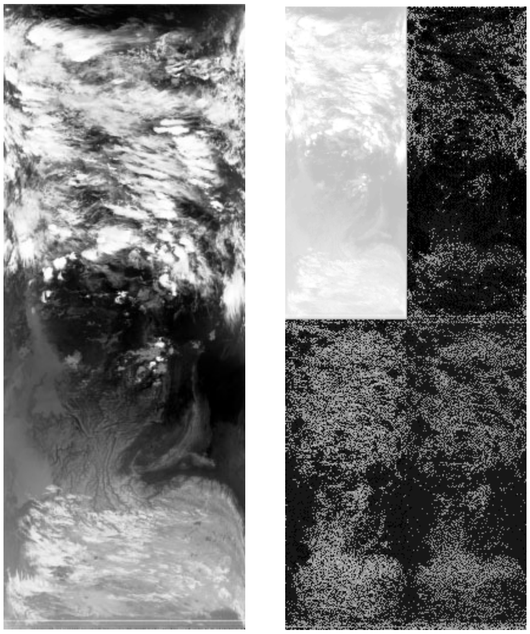
2. Then a client requests information to recover higher resolution data and Block 1 ($L = 1$) coefficients are sent back to a client. A user agent at a client machine receives Block 1 and reconstruct upper right quadrant of the image. The resolution then increases by a factor of 2.

3. If a client further requests higher resolution data to have a higher resolution image of the upper right corner, then Block 3 ($L = 2$) and Block 7 ($L = 3$) is subsequently sent to a client, and a user agent at the client side reconstruct the original image subsequently. After the receipt of the Block 7, the client will have perfectly reconstructed original image of the upper right corner with the size of 1/16 to the original image.

5 Experiment

Wavelet transform We apply the wavelet transform stated above, and see how a real satellite image is transformed to wavelet coefficients. Satellite images used for this experiments are NOAA-AVHRR satellite images (size 2048×5296) received at the Asian Center for Research on Remote Sensing (ACRoRS) in Asian Institute of Technology (AIT), Thailand. As addressed earlier, these data are being exchanged between AIT and academic institutions in Japan using NACSIS-Thai International Link.

Entropy of sub blocks The original image and transformed image are shown in Figure 4. This is a



(a) Original (b) Wavelet transformed

Figure 4: Wavelet (S+P) transform applied to NOAA satellite imagery.

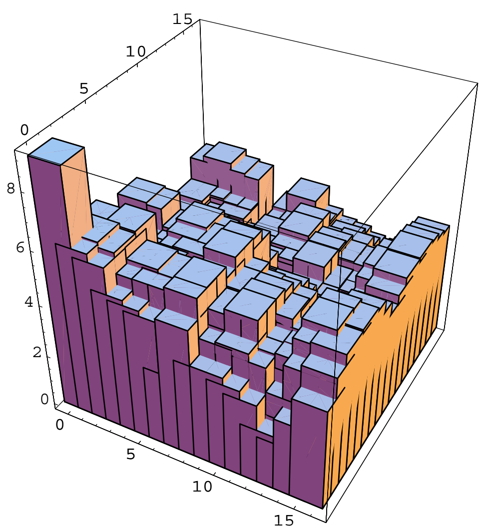


Figure 5: Entropy of sub blocks (16×16). Note that the original satellite image has 10 bits.

basic process of pyramidal algorithm. In the upper left corner, low resolution image is produced, while for other quadrants, edges or points are remained. Combining those four quadrants will perfectly reconstruct the original image.

Next we further advance wavelet transform and decompose the same image into $16 \times 16 = 256$ sub blocks of size 128×331. Entropy for each sub block is shown in Figure 5. The most low resolution sub block has in particular higher entropy, and lower resolution blocks shows relatively smaller entropy compared to higher resolution blocks.

Preliminary discussion If a client knows beforehand which area in an image he / she wants to observe, then the method with minimal transmission of data is to send directly only the required area from a client. Compared to this method, experimental results indicates that the total amount of data transmitted to a user is about 4 times bigger than the method above (specific numbers not shown here) . Taking an example in the above scenario, this inefficiency is caused because if a user wants to have Block 7, a user must receive other blocks 0, 1, 3, which contains data on unnecessary regions, such as lower left corner contained in the Block 0 image.

However, if a client wants to browse the image in the first place, progressive transmission-based scheme works effective because a client side keeps rough data of the whole image. Quantitative comparison based on many kinds of access scenario should be carried out in the future to verify in which situation our proposed schemes work effectively.

6 Conclusion

In this paper, we presented several results related to multiresolution cache management. Access skew on satellite data can be exploited to improve the efficiency of the caching server specially designed for distributed satellite image databases. Multiresolution image representation, which is based on the wavelet-based image compression methods, is a promising research direction for a new caching mechanism, however we cannot show many results on this part of research.

Future works include consideration on prefetch. If a user is accessing low resolution data, it is highly likely that a user may access to higher resolution data of the same image. Then it may be effective to predict the access pattern of the individual user and improve the efficiency of the caching server.

Acknowledgement

The author thanks Prof. Kitsuregawa and Dr. Nemoto at Institute of Industrial Science (IIS), University of

Tokyo, for providing access pattern of GMS satellite images used in this paper.

References

[1] C. Aggarwal, J. Wolf, and P. Yu. Caching on the World Wide Web. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):94–107, 1999.

[2] P. Barford, A. Bestavros, A. Bradley, and M. Crovella. Changes in web client access patterns: Characteristics and caching implications. *World Wide Web Journal*, 2:15–28, 1999.

[3] P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *IEEE Trans. Commun.*, 31(4):532–540, 1983.

[4] B. Huberman, P. Pirolli, J. Pitkow, and R. Lukose. Strong regularities in World Wide Web surfing. *Science*, 280(3):95–97, 1998.

[5] T. Nemoto and M. Kitsuregawa. Load balancing algorithm using tape migration mechanisms for scalable tape archiver and its performance evaluation. *The Transactions of the Institute of Electronics, Information, and Communication Engineers*, J82-D-I(1):53–69, 1999. (Japanese).

[6] M. Oguchi. A study on caching proxy mechanisms in a wide area network envionment. *Research Bulletin of NACSIS*, 8, 1996.

[7] M. Oguchi and K. Ono. A study of caching proxy mechanisms realized on wide area distributed networks. In *Proc. of the Fifth IEEE International Symposium on High Performance Distributed Computing*, pages 443–449, 1996.

[8] A. Said and W. Pearlman. An image multiresolution representation for lossless and lossy compression. *IEEE Transactions on Image Processing*, 5(9):1303–1310, 1996.

[9] J. Shim, P. Scheuermann, and R. Vingralek. Proxy cache algorithms: Design, implementation, and performance. *IEEE Transactions on Knowledge and Data Engineering*, 11(4):549–562, 1999.

[10] R. Tewari, H. Vin, A. Dan, and D. Sitaram. Resource based caching for web servers. In *Proceedings of SPIE/ACM Conference on Multimedia Computing and Networking*, 1998.

[11] G. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 3(4):30–44, 1991.

[12] I. Witten, R. Neal, and J. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, 1987.

[13] Y. Yasuda, M. Takagi, S. Kato, and T. Awano. Step by step image transmission and display from gross to fine information using hierarchical coding. *The Transactions of the Institute of Electronics and Communication Engineers*, J63-B(4):379–386, 1980.

[14] Y. Yasuda, T. Yasuno, F. Katayama, T. Toida, and H. Sakata. Image database system featuring graceful oblivion. *The Transactions of the Institute of Electronics, Information, and Communication Engineers*, E79-B(8):1015–1022, 1996.