COMPUTATIONALLY EFFICIENT ALGORITHMS FOR PREDICTING THE FILE SIZE OF JPEG IMAGES SUBJECT TO CHANGES OF QUALITY FACTOR AND SCALING

*Steven Pigeon and Stéphane Coulombe**

École de technologie supérieure
Department of Software and IT Engineering
1100 Notre Dame Ouest, Montréal, Qc, Canada, H3C 1K3
e-mail: steven.pigeon@etsmtl.ca, stephane.coulombe@etsmtl.ca

ABSTRACT

To enable the delivery of multimedia content to mobile devices with limited capabilities, high volume transcoding servers must rely on efficient adaptation algorithms. Our objective in addressing the case of JPEG image adaptation was to find computationally efficient algorithms to accurately predict the compressed file size of images subject to simultaneous changes in quality factor (QF) and resolution. In this paper, we present two new prediction algorithms which use only information readily available from the file header. The first algorithm, QF Scaling-Aware Prediction, predicts file size based on the QF of the original picture, as well as a target QF and scaling. The second algorithm, Clustered QF Scaling-Aware Prediction, also takes into account the resolution of the original picture for improved prediction accuracy. As both algorithms rely on machine-learning strategies, a large corpus of representative JPEG images was assembled. We show that both prediction algorithms lead to acceptably small relative prediction errors in adaptation scenarios of interest.

1. INTRODUCTION

It is an undeniable fact that the heterogeneous nature of mobile terminals renders multimedia adaptation inevitable. For instance, with Multimedia Messaging Services (MMS), server-side message adaptation is necessary to ensure interoperability [1]. As a large proportion of the media adaptations in multimedia messaging involves JPEG images (generated by an ever-increasing number of camera phones), our paper focuses on that particular case. In this context, interoperability problems are mainly due to image resolution or a file size exceeding the receiving device's capabilities. Indeed, the small memory footprint of such devices limits the maximum file size and resolution of the images they accept. A great deal of research has focused on the problem of efficiently reducing the resolution of images (see [2] for details and more references). However, efficiently reducing an image's compressed file size to a given target, especially while maximizing the perceived quality of the image, remains a challenge.

In the lossy JPEG image format, the user controls file size via a quality factor (QF) which affects the quantization process, and ultimately the file size itself [3]. A higher QF leads to better image quality and a larger file. Because it also depends on other image properties, the precise relationship between the QF and the compressed file size is still not known. A simple transcoding approach to reducing file size might be to decode the image and iteratively re-encode it with different QF values until the target size is met (within an acceptable tolerance). Although functional, this method would be highly inefficient in terms of computation effort, and would not be acceptable for implementation in high volume image transcoding servers.

Accordingly, several studies have investigated the relationship between quantization and compressed file size (or bit rate) [4–9]. Although they provide valid and interesting results, these studies might not be applicable to our problem because many of their assumptions do not hold or are too restrictive. Also, some of these studies were conducted in the context of H.263 or MPEG video coding, which use simpler quantization models than JPEG. Finally, they require the computation of complex model parameters or pixel-level image statistics, which are quite computation-intensive. Ridge addresses the specific problem of JPEG size adaptation and presents a reliable method for achieving file size reduction [2]. However, his solution requires that statistics at the coded image syntax level be gathered (e.g. number of zeroed DCT coefficients). This not only increases the complexity of the process, but requires some level of re-engineering of the image compression tools, as the JPEG encoder/decoder software has to become a specialized transcoder. Another drawback of the above-mentioned methods is that they consider that the resolution is fixed (or altered independently in a previous stage) and focus solely on file size reduction through changes in quantization, while we

*This work was funded by Vantrix Corp. and by the Natural Science and Engineering Research Council of Canada, under the Collaborative Research and Development Program (NSERC-CRD 326637-05). The authors wish to thank Jean-François Franche for contributing to some of the ideas and participating in the preliminary investigations on which this paper is based.

978-1-4244-1946-3/08/$25.00

believe that image scaling can be used as a complementary strategy to achieve that reduction.

In this paper, we present two computationally efficient algorithms to predict the compressed file size of a JPEG image subject to changes to both its QF and resolution (scaling). Such prediction algorithms could be implemented in a transcoder to determine appropriate values of QF and scaling parameters leading to a target image file size. How to best select the combination of QF and scaling that will maximize perceptual quality while meeting a size constraint is the topic of an upcoming paper.

2. PROBLEM STATEMENT

In this section, we formally define the problem of image adaptation to meet a device's capabilities. If I is a JPEG compressed image, let $QF(I)$ be the QF used to create it, $S(I)$ its compressed file size, and $W(I)$ and $H(I)$ its width and height in pixels respectively. Here, and in the remainder of this paper, we assume that the semantics of the QF comply with the Independent JPEG Group definition, where $1 \leqslant QF \leqslant 100$, from coarsely quantized to essentially lossless [3]. Similarly, for a receiving terminal device D, let $W(D)$, $H(D)$, and $S(D)$ be the maximum image width, height, and compressed file size supported respectively.

The compressed image file size may be adapted (or transcoded) by altering the QF or through scaling, or both, in order to comply with the limited capabilities of D. This operation is typically performed on the server side (e.g. in a gateway). The image is resized using an aspect-preserving scaling, or *zoom*, factor $0 < z \leqslant 1$. A JPEG transcoding operation, denoted $T(I, QF_{out}, z)$, is the function that returns the compressed JPEG image resulting from the application, to the image I, of both the new QF (QF_{out}) and the scaling factor z. A JPEG transcoding operation $T(I, QF_{out}, z)$ is defined as *feasible* for the device D, if, for parameters I, $1 \leqslant QF_{out} \leqslant 100$, and $0 < z \leqslant 1$, it meets the following constraints:

$$S\big(\ T(I, QF_{out}, z)\ \big) \leqslant S(D)$$

$$W\big(\ T(I, QF_{out}, z)\ \big) \leqslant W(D)$$

$$H\big(\ T(I, QF_{out}, z)\ \big) \leqslant H(D)$$

Since there are likely many combinations of QF_{out} and z that lead to feasible transcodings, an optimization algorithm will attempt to find the optimal values, $QF^*_{out}(I)$ and $z^*(I)$, for image I and device D, that minimize a certain metric. For instance, the metric could be the difference between $S\big(T(I, QF_{out}, z)\big)$ and $S(D)$ (to select the transcoded image, the size of which is closest to the maximum allowed size), or a measure of the perceptual quality of the transcoded image, etc. The choice of such a metric is beyond the scope of this paper.

An optimization algorithm searching for optimal values $QF^*_{out}(I)$ and $z^*(I)$ must compute $S\big(\ T(I, QF_{out}, z)\ \big)$ for a number of possible values of QF_{out} and z, which is potentially very expensive. We propose using a computationally inexpensive predictor $\hat{S}(I, QF_{out}, z)$, rather than the exact function $S\big(\ T(I, QF_{out}, z)\ \big)$ using actual transcodings. We add the constraint that the predictor must compute its prediction using only readily available information about I, such as $S(I)$, $W(I)$, $H(I)$, and $QF(I)$, thereby avoiding any costly pixel-level or compressed domain processing. The prediction algorithm for $\hat{S}$ must also be accurate, so that an optimization algorithm (which is beyond the scope of this paper, as we are concentrating on the prediction problem only) can reliably use it to perform efficient adaptation.

3. PROPOSED PREDICTION ALGORITHMS

In this section, we propose two prediction algorithms based on machine-learning techniques. First, for both predictors, a transcoded image file size model is proposed. Then, the predictors undergo a *training phase*, where numerous exemplars of images with various transcoding parameters and a known transcoded file size are used to optimize the predictors. Finally, the accuracy of the predictors is verified through a *test phase*, where the model is presented with new exemplars and the predictions compared to the known solutions.

The first algorithm, *QF Scaling-Aware Prediction*, uses only the original image QF (QF_{in}) and the desired output QF (QF_{out}) and scaling (z) to formulate its prediction. The second algorithm, *Clustered QF Scaling-Aware Prediction*, refines the first by using the original resolution of the image as well. This allows the algorithm to refine its prediction for classes of images of similar resolution, thereby enhancing the prediction accuracy significantly.

3.1. The Image Corpus and Training Methodology

Optimizing and testing the proposed prediction algorithms require an image corpus. Unfortunately, a large database of typical JPEG images sampled from multimedia applications was not available to us. Therefore, we developed a crawler for the extraction of images from popular Web sites. The corpus we assembled contains about 70,300 JPEG files. It is free of corrupted files and all meta-data (EXIF) were removed. For each image I in the corpus, a large number of transformations was applied using different QF_{out} and z ($QF_{out} = 10, 20, \ldots, 100$, $z = 0.1, 0.2, \ldots, 1.0$), and the resulting file size was recorded. For each transcoding, we formed the vector $\big(I, QF(I), W(I), H(I), S(I), QF_{out}, z, S(T(I, QF_{out}, z))\big)$. Let all these vectors form the augmented image corpus, denoted C. A random partition of C into two disjoint sets, in an 80/20 proportion, forms the training set T and the test set Q respectively. The transcodings were generated using ImageMagick's command-line tools, version 6.2.4 [10] and the Blackman filter for scaling.

3.2. QF Scaling-Aware Prediction

This first algorithm predicts the compressed file size of the transcoded picture following application of a new QF $1 \leqslant QF_{out} \leqslant 100$ and scaling factor $0 < z \leqslant 1$. The predictor, denoted $\hat{S}\big(I, QF_{out}, z\big)$, is given by

$$\hat{S}\big(I, QF_{out}, z\big) = S(I)\hat{s}\big(QF(I), QF_{out}, z\big) \tag{1}$$

The function $\hat{s}$ is a relative size predictor given by

$$\hat{s}\big(QF(I), QF_{out}, z\big) = \frac{1}{|\mathsf{T}_{QF(I)}|} \sum_{J \in \mathsf{T}_{QF(I)}} s\big(J, QF_{out}, z\big) \tag{2}$$

where $s(J, QF_{out}, z)$ is the exact function

$$s\big(J, QF_{out}, z\big) = \frac{S\big(\,T(J, QF_{out}, z)\,\big)}{S\big(J\big)}$$

Here, $\mathsf{T}_{QF(I)} \subseteq \mathsf{T}$ is the subset of images in the training set T of the same QF as I, $|\mathsf{T}_{QF(I)}|$ is its cardinality, and $T(J, QF_{out}, z)$ is the function that returns the compressed image resulting from the application, to J, of both the new QF (QF_{out}) and scaling factor z. In simpler terms, using each image in T having the same QF as the original image and the same QF_{out} and z values as the transcoding to apply, we compute the average ratio between the transcoded image file size and the original image size. It is important to note that $\hat{s}$ is an optimal least mean squares estimator.

As the function $\hat{s}$ is expensive to compute, it should be precomputed into an array M, the indices of which are the *quantized* original QF (QF_{in}), the transcoded QF (QF_{out}), and the scaling factor z. Here, tilde $\sim$ denotes quantized values. Let $\widetilde{QF}_{in}$ be the quantized input QF, $\widetilde{QF}_{out}$ the quantized desired output QF, and $\tilde{z}$ the quantized scaling. In our experiments, we used the quantized values $\{10, 20, \ldots, 100\}$ for $\widetilde{QF}_{in}$ and $\widetilde{QF}_{out}$ and $\{0.1, 0.2, \ldots, 1.0\}$ for $\tilde{z}$. According to this scheme, the relative size prediction for quantized input $\widetilde{QF}_{in}$, quantized desired output $\widetilde{QF}_{out}$, and quantized scaling $\tilde{z}$ is given by:

$$M_{\widetilde{QF}_{in}, \widetilde{QF}_{out}, \tilde{z}} = \big|\mathsf{S}_{\widetilde{QF}_{in}, \widetilde{QF}_{out}, \tilde{z}}\big|^{-1} \sum_{t \in \mathsf{S}_{\widetilde{QF}_{in}, \widetilde{QF}_{out}, \tilde{z}}} \hat{s}\big(QF_{in}(t), QF_{out}(t), z(t)\big) \tag{3}$$

where $\mathsf{S}_{\widetilde{QF}_{in}, \widetilde{QF}_{out}, \tilde{z}}$ is the set of all transformed images, the parameters of which fall in the quantization cells $\widetilde{QF}_{in}$, $\widetilde{QF}_{out}$, and $\tilde{z}$, and where $QF_{in}(t)$ returns the original QF of transformed image t, $QF_{out}(t)$ returns the output QF, and $z(t)$ the scaling that were applied. The function $\hat{s}$ is given by eq. (2). Accordingly, $M_{\widetilde{QF}_{in}}$ denotes a slice of that array, a matrix, with indices $\widetilde{QF}_{out}$ and $\tilde{z}$. The prediction of (1) now becomes:

$$\hat{S}\big(I, QF_{out}, z\big) = S(I)\; M_{\widetilde{QF}_{in}, \widetilde{QF}_{out}, \tilde{z}} \tag{4}$$

The quantization scheme is not fixed by this algorithm. The user can choose the quantization scheme that best matches his expected traffic. A suitably coarse quantization will prevent *context dilution*, a situation that occurs when the number of exemplars corresponding to a given context (here, $\big(\widetilde{QF}_{in}, \widetilde{QF}_{out}, \tilde{z}\big)$ forms the context) is insufficient to draw reliable statistics. It is interesting to note that, although the model proposed in eq. (1) (as well as in eq. (4)) does not use explicit statistics related to the compressed form of the input image (such as the number of zeroed DCT coefficients), it implicitly takes into account the compressibility of the input image through its file size, $S(I)$.

3.3. Clustered QF Scaling-Aware Prediction

This second prediction algorithm is based on the prediction model of the first algorithm. However, it refines that algorithm by using the original image resolution as well, in order to alleviate the effect of outliers from which the first algorithm suffers (see section 4.1). It extends the prediction's input parameters from $\big(QF_{in}, QF_{out}, z\big)$ to $\big(QF_{in}, H_{in}, W_{in}, QF_{out}, z\big)$. However, the distribution of the input heights (H_{in}) and widths (W_{in}) would cause context dilution, unless they were quantized. The method uses *clustering* to overcome this problem. Clustering is a technique which partitions data in a given number of disjoint subsets, *classes*, so that data in each subset are maximally similar under the chosen metric. For each subset, a representative value, or *prototype*, is computed, in our case, the centroid.

To each image I in the training set T, we associate a vector $x_I = \big(W(I), H(I), \alpha\, QF(I)\big)$, where α is a constant to bring the QF dimension to the same order of magnitude as the width and height dimensions; it is necessary to do so because the error measure for the clustering is the L_2 norm. We have found empirically that $\alpha \approx 1000$ gives good results. The number of classes, k, will also be chosen prior to clustering. The parameter k has to be large enough to reduce the error, and yet small enough to avoid context dilution. In our experiments, we set $k = 200$.

We must compute a partition P of the training set T into k subsets. By definition, the partition $\mathsf{P} = \{\mathsf{P}_1, \mathsf{P}_2, \ldots, \mathsf{P}_k\}$ must satisfy $\bigcup_{i=1}^{k} \mathsf{P}_i = \mathsf{T}$ and $\bigcap_{i=1}^{k} \mathsf{P}_i = \varnothing$. The optimal partition P^* minimizes the expected squared distance between any vector x_I (with $I \in \mathsf{T}$) and its assigned centroid, that is,

$$\mathsf{P}^* = \arg\min_{\mathsf{P}} \sum_{i=1}^{k} \sum_{I \in \mathsf{P}_i} \|x_I - \bar{x}_i\|^2$$

where $\|x\| = \sqrt{x^T x}$ is the L_2 norm, and $\bar{x}_i$ is the centroid of class P_i given by:

$$\bar{x}_i = \frac{1}{|\mathsf{P}_i|} \sum_{I \,\in\, \mathsf{P}_i} x_I$$

The optimal partition P^* cannot be exactly computed in any reasonable length of time, but it can be approximated using the k-means algorithm [11]. Once the partition P is computed from the training set, we create, for each centroid $\bar{x}_i$, a prediction matrix $M_{\bar{x}_i}$. Each of these matrices has two dimensions, the quantized QF and the quantized scaling factors. The entries $M_{\bar{x}_i,\widetilde{QF}_{out},\tilde{z}}$ are computed in a similar way to eq. (2):

$$M_{\bar{x}_i,\widetilde{QF}_{out},\tilde{z}} = \frac{1}{|\mathsf{P}_i|} \sum_{J \,\in\, \mathsf{P}_i} \frac{S\big(T(J,\widetilde{QF}_{out},\tilde{z})\big)}{S(J)} \qquad (5)$$

where $J \in \mathsf{P}_i$ is an image which was assigned to the class P_i, with centroid $\bar{x}_i$, and of cardinality $|\mathsf{P}_i|$. To find the predictor associated with an image J, we find the closest centroid $\bar{x}_J$:

$$\bar{x}_J = \underset{\bar{x}_i \,\in\, \mathsf{P}}{\arg\min}\ \|x_J - \bar{x}_i\|$$

The final prediction is

$$\hat{S}\big(I,\widetilde{QF}_{out},\tilde{z}\big) = S(I)\ M_{\bar{x}_I,\widetilde{QF}_{out},\tilde{z}}$$

The cost of this prediction algorithm is limited to the cost of finding the closest centroid that can be computed efficiently. Reading from the matrix $M_{\bar{x}_J}$ can be achieved in constant time. The initial computation of the centroids is more expensive, although the k-means algorithm is quite efficient. The cost of the main loop of the k-means algorithm is $O(n \log k)$, where $n = |\mathsf{T}|$ and k is the number of classes. The number of iterations needed for convergence varies, but the magnitude of the relative error is known to decrease very rapidly [12, sect. 5]. Our experiments also confirm this result, showing that the number of iterations needed rarely exceeds 50.

4. EXPERIMENTAL RESULTS

4.1. Results for QF Scaling-Aware Prediction

As described in section 3.2, we computed the array M, indexed by quantized indices $\widetilde{QF}_{in}$, $\widetilde{QF}_{out}$, and $\tilde{z}$, using eq. (3) and all the exemplars from the training set. The array $M_{\widetilde{80}}$, corresponding to a slice of array M with $\widetilde{QF}_{in} = 80$, is shown in Table 1, where we see the file size scaling ratio expected for each value of QF and scaling.

In order to validate the prediction algorithm, we have computed the expected relative absolute error, $\mathrm{E}[|S(I_{out}) - \hat{S}(I_{out})|/S(I_{out})] \times 100\%$, obtained from transcoding pictures from the test set Q having $\widetilde{QF}_{in} = 80$ to form I_{out} for various values of $QF_{out} = \{10, 20, \ldots, 100\}$ and scaling factors $z = \{10\%, 20\%, \ldots, 100\%\}$. The results are

$\widetilde{QF}_{out}$ \ Scaling, $\tilde{z}$	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
10	0.03	0.04	0.05	0.07	0.08	0.10	0.12	0.15	0.17	0.20
20	0.03	0.05	0.07	0.09	0.12	0.15	0.19	0.22	0.26	0.32
30	0.04	0.05	0.08	0.11	0.15	0.19	0.24	0.29	0.34	0.41
40	0.04	0.06	0.09	0.13	0.17	0.22	0.28	0.34	0.40	0.50
50	0.04	0.06	0.10	0.14	0.19	0.25	0.32	0.39	0.46	0.54
60	0.04	0.07	0.11	0.16	0.22	0.28	0.36	0.44	0.53	0.71
70	0.04	0.08	0.13	0.18	0.25	0.33	0.42	0.52	0.63	0.85
80	0.05	0.09	0.15	0.22	0.31	0.41	0.52	0.65	0.78	0.95
90	0.06	0.12	0.21	0.31	0.44	0.59	0.75	0.93	1.12	1.12
100	0.10	0.24	0.47	0.75	1.05	1.46	1.89	2.34	2.86	2.22

Table 1. The matrix $M_{\widetilde{80}}$, optimized from the image training set described in section 3.1 with QF Scaling-Aware Prediction.

$\widetilde{QF}_{out}$ \ Scaling, $\tilde{z}$	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
10	112.9	69.63	48.51	36.74	28.96	24.75	21.36	18.90	17.22	15.70
20	92.75	52.81	35.78	26.65	20.53	17.52	14.93	12.97	11.63	10.23
30	82.23	44.89	30.09	22.07	16.77	14.22	11.90	10.22	8.92	7.55
40	75.74	40.34	26.84	19.52	14.64	12.32	10.15	8.57	7.27	6.45
50	70.74	36.99	24.49	17.70	13.11	10.96	8.88	7.36	6.04	6.32
60	66.28	34.14	22.48	16.19	11.82	9.84	7.81	6.36	5.00	2.40
70	60.75	30.69	20.14	14.46	10.42	8.57	6.61	5.30	4.05	2.40
80	54.08	26.83	17.56	12.65	8.97	7.33	5.55	4.50	3.53	2.42
90	44.44	21.69	14.64	10.83	7.89	6.72	5.72	5.22	4.89	2.88
100	28.84	18.59	16.62	16.17	15.39	15.01	14.70	14.06	13.90	8.39

Table 2. The expected relative absolute error $\mathrm{E}[|S(I_{out}) - \hat{S}(I_{out})|/S(I_{out})] \times 100\%$, of the QF Scaling Aware Prediction, for matrix $M_{\widetilde{80}}$.

shown in Table 2. As expected, the error is minimal around $\widetilde{QF}_{in} = \widetilde{QF}_{out} = 80$ with $\tilde{z} = 100\%$. The error grows as $\widetilde{QF}_{out}$ and $\tilde{z}$ differ more and more from the input. Table 3 gives the probabilities that the absolute relative error is under a certain threshold β for a typical $\widetilde{QF}_{in} = \widetilde{QF}_{out} = 80$, that is, $P\big(\ |S(I_{out}) - \hat{S}(I_{out})| < \beta\, S(I_{out})\ \big|\ \tilde{z},\ \widetilde{QF}_{in} = 80,\ \widetilde{QF}_{out} = 80\ \big)$ for different $\tilde{z}$ and β. The distribution of the error spreads as we move further away from scalings of 100%, as expected.

Overall, the algorithm is very simple to implement and requires very little processing once the prediction tables have been precomputed. The relative prediction error is reasonably small for values of QF and scaling close enough to those of the original image. However, it becomes increasingly imprecise as we move further away from the original image's properties. Also, the algorithm is sensitive to outliers such as small images for which the header size is not negligible compared to the overall image size.

4.2. Results for Clustered QF Scaling-Aware Prediction

Because it is not easy to visually represent the numerous clusters generated by this algorithm for the training set T, and to make it easier to compare it with the first algorithm, the absolute relative error results of many clusters were cumulated ignoring resolution. The results are shown in Table 4. As with the previous algorithm, the minimal errors are

$$P(\,|S(I_{out}) - \hat{S}(I_{out})| < \beta\, S(I_{out}))$$

β	Scaling 10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
0.1	0.09	0.22	0.36	0.49	0.66	0.75	0.87	0.92	0.94	0.97
0.2	0.20	0.48	0.68	0.82	0.92	0.96	0.98	0.99	1.00	1.00
0.3	0.36	0.68	0.85	0.94	0.97	1.00	1.00	1.00	1.00	1.00

Table 3. The probability that the absolute relative error is under a certain threshold β, for QF Scaling-Aware Prediction, with $\widetilde{QF}_{in} = 80$ and $\widetilde{QF}_{out} = 80$.

$\widetilde{QF}_{out}$	Scaling, $\tilde{z}$ 10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
10	24.82	21.84	19.47	17.75	16.20	14.95	14.06	13.37	12.87	12.71
20	23.80	20.25	17.38	15.32	13.41	12.03	10.93	9.99	9.32	8.86
30	23.17	19.28	16.26	14.06	12.00	10.55	9.32	8.29	7.51	6.95
40	22.76	18.65	15.53	13.27	11.10	9.61	8.35	7.25	6.37	6.04
50	22.43	18.14	14.98	12.67	10.48	8.93	7.60	6.46	5.51	5.97
60	22.11	17.69	14.42	12.10	9.87	8.31	6.93	5.74	4.73	2.18
70	21.65	17.11	13.79	11.45	9.18	7.57	6.18	4.95	3.93	1.99
80	21.12	16.41	13.10	10.70	8.39	6.79	5.38	4.23	3.31	1.89
90	20.42	15.67	12.43	10.08	7.79	6.44	5.28	4.45	3.82	2.19
100	20.86	18.20	16.22	15.06	13.48	12.99	12.34	11.58	11.13	6.53

Table 4. Expected absolute relative error, $\times 100\%$, for clustered prediction matrix $M_{\bar{x}_I}$ for pictures with $\widetilde{QF}_{in} = 80$.

concentrated around $\widetilde{QF}_{in} = \widetilde{QF}_{out} = 80$ with $\tilde{z} = 100\%$. However, the region of expected absolute errors of 10% or less grew significantly relative to those in Table 2, and the maximal errors were greatly reduced — from 112.9% to 24.8% for the most imprecise prediction case shown. We can also see clear improvements in Table 5 compared to Table 3. The increased complexity of the second method is therefore entirely justified by the improved prediction accuracy.

5. CONCLUSION

This paper addressed the problem of predicting the file size of an image subject to a simultaneous change in quality factor and resolution. Two prediction algorithms were proposed: QF Scaling-Aware Prediction and Clustered QF Scaling-Aware Prediction. The latter provides a significantly better prediction accuracy at a moderate increase in computational cost. Still, both algorithms are simple to implement and computationally efficient, and both are therefore ideally suited for high volume transcoding servers.

$$P(\,|S(I_{out}) - \hat{S}(I_{out})| < \beta\, S(I_{out})\,)$$

β	Scaling, $\tilde{z}$ 10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
0.1	0.33	0.41	0.49	0.57	0.68	0.78	0.86	0.93	0.96	0.98
0.2	0.60	0.70	0.80	0.88	0.94	0.97	0.98	0.99	1.00	1.00
0.3	0.77	0.87	0.96	0.96	0.99	0.99	1.00	1.00	1.00	1.00

Table 5. The probability that the absolute relative error is under a certain threshold β, for $\widetilde{QF}_{in} = 80$ and $\widetilde{QF}_{out} = 80$ with clustered QF Scaling-Aware Prediction.

6. REFERENCES

[1] S. Coulombe and G. Grassel, "Multimedia adaptation for the multimedia messaging service," *IEEE Communications Magazine*, vol. 42, no. 7, pp. 120–126, July 2004.

[2] J. Ridge, "Efficient transform-domain size and resolution reduction of images," *Signal Processing: Image Communication*, vol. 18, no. 8, pp. 621–639, Sept. 2003.

[3] T. Lane, P. Gladstone, L. Ortiz, J. Boucher, L. Crocker, J. Minguillon, G. Phillips, D. Rossi, and G. Weijers, "The independent jpeg group software release 6b," 1998, ftp://fp.uu.net/graphics/jpeg/jpegsrc.v6b.tar.gz.

[4] P. A. A. Assunção and M. Ghanbari, "A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit streams," *IEEE Trans. Circuits and Systems For Video Technology*, vol. 8, no. 8, pp. 953–967, Dec. 1998.

[5] H. Sorial, W. E. Lynch, and A. Vincent, "Selective requantization for transcoding of MPEG compressed video," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2000, vol. 1, pp. 217–220.

[6] S. W. Wu and A. Gersho, "Rate-constrained picture-adaptive quantization for JPEG baseline coders," in *The 1993 International Conference on Acoustics, Speech, and Signal Processing*, Apr. 1993, vol. 5, pp. 389–392.

[7] G. S. Yovanof and A. I. Drukarev, "Fixed rate JPEG compliant still image compression," Oct. 1997, US Patent 5,677,689.

[8] Z. He, Y. K. Kim, and S. Mitra, "Low-delay rate-control for DCT video coding via ρ-domain source modeling," *IEEE Trans. Circuits and Systems For Video Technology*, vol. 11, no. 8, pp. 928–940, Aug. 2001.

[9] L-J Lin, A. Orthega, and C. C. J. Kuo, "Rate control using spline-interpolated R-D characteristics," in *Procs. VCIP'96*, 1996, pp. 111–122.

[10] "ImageMagick command line tools," http://www.imagemagick.org/script/index.php.

[11] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Comm.*, vol. 28, no. 1, pp. 84–95, 1980.

[12] L. Bottou and Y. Bengio, "Convergence properties of the K-means algorithms," in *Advances in Neural Information Processing Systems*, G. Tesauro, D. Touretzky, and T. Leen, Eds. 1995, vol. 7, pp. 585–592, The MIT Press.