Efficient Clustering-based Algorithm for Predicting File Size and Structural Similarity of Transcoded JPEG Images

Steven Pigeon and Stéphane Coulombe
Department of Software and Information Technology Engineering
École de Technologie Supérieure
1100 Notre-Dame West, Montréal, Québec, H3C 1K3

January 23, 2012

Abstract

The problem of adapting JPEG images to satisfy constraints such as file size and resolution arises in a number of applications, from universal media access to multimedia messaging services. Visually optimized adaptation, however, commands a non-negligible computational cost which we aim to minimize using predictors. In previous works, we presented predictors and systems to achieve low-cost near-optimal adaptation of JPEG images. In this work, we propose a new approach to file size and quality prediction resulting from the transcoding of a JPEG image subject to changes in quality factor and resolution. We show that the new predictor significantly outperforms the previously proposed solutions in accuracy.

1 Introduction

The need for efficient image adaptation arises in a number of contexts, ranging from universal media access with varying browsing conditions [1, 2] to multimedia messaging services (MMS) [3], and in each case, the challenge is to adapt images to fit given receiving terminal and application constraints while simultaneously maximizing the user experience and minimizing the computational cost of adaptation.

In the context of MMS, for example, a receiving terminal is characterized by its capabilities—or more exactly its limitations—such as the maximum resolution of images it can display and the maximum message size it can receive and interpret correctly [4]. Adaptation can also include the case where the compression format itself needs to be changed, but observations show that MMS image traffic, dominated by camera phone images, is mostly composed of JPEG images. Accordingly, we will, in this study, concentrate on the case of JPEG to JPEG image adaptation subject to changes in quality factor and scaling.

Previous studies have addressed this particular problem, but the solutions they propose are either still computationally expensive (and necessitating extensive modifications to existing JPEG library software) or overly rigid, focusing on unrealistically constrained transformations such as scaling by powers of two [5–7], or using only a small, fixed, number of possible adaptations, without consideration for the perceived adaptation quality [1,8]. In the context of high-volume service providing, whether for MMS or universal media access, only the fastest adaptation methods yielding the best perceived quality can be considered.

For this purpose, we have, in previous works, proposed low-cost predictor-based adaptation systems [9–11]. These constant-time predictors, which we will refer to as JQSP1 (for JPEG Quality and Size Predictor) and JQSP2 in this work, are described in section 3. The predictors are used in combination with an adaptation system to predict the best transcoding parameters for adapting a JPEG image to given terminal constraints, where "best" is defined as most likely to minimize perceived distortion under the considered viewing conditions as defined by the characteristics of the receiving terminal.

These predictors, although shown to perform satisfactorily, did not make use of all the available information about images such as bits per pixel and resolution, both of which are very likely to help formulate more accurate predictions about file size and quality resulting from a given transcoding operation. However, the table-based schemes in our earlier works do not easily lend themselves to a larger number of parameters, and we believe that the uniform quantization of parameters is also an unnecessary restriction for the problem at hand.

In this work, we propose to extend and improve the solutions previously presented by the authors by including more information about images to in order help formulate more accurate predictions on file size and quality resulting from transcoding, and by lifting the restrictions of the previous methods, in particular the uniform quantization of parameters. To do so, we propose a method based on the clustering of transcoding operations as high-dimensional vectors.

This work is structured as follows. The next section, section 2, presents the proposed solution. Section 3 details the test methodology as well as the algorithms presented in previous works. Section 4 presents the results from the proposed and previously proposed prediction methods. In section 5, we discuss the results, accuracy, algorithmic complexity, and the memory usage of the algorithms considered. Lastly, section 6 summarizes the results and presents our concluding remarks.

2 Proposed Clustering Solution

In this section, we present the details of the enhanced predictor, EJQSP, the solution we are proposing for the enhanced prediction of relative file size and quality resulting from a JPEG image adaptation based on clustering [12]. We first describe the general problem of clustering, then we describe its application to our particular objective.

The general clustering problem is as follows. We have n vectors in a d-dimensional space that we wish to segregate into m different classes, that is, create a partition. Each subset of the partition is represented by a single vector, its prototype, chosen so that the average distance (or error) between the exemplars in the subset and the prototype is minimized under a given metric. The objective is to find the partition with m subsets that minimizes the overall error on all vectors.

More formally, let $X=\{x_1,x_2,\ldots,x_n\}$ be the n exemplars in $\mathbb{R}^d$, which we want to partition into $C=\{C_1,C_2,\ldots,C_m\}$, the m subsets. The partition C is such that $\bigcup_{i=1}^m C_i=X$ and $C_i \cap C_j=\varnothing$, for $1 \leqslant i \neq j \leqslant m$. Since the metric considered is the Euclidean distance, the prototype for a subset C_i is its centroid, given by

$$\bar{x}_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j \ ,$$

and let $\bar{X} = \{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m\}$ be the centroids corresponding to the subsets. The error associated with a partition C is

$$E(C) = \sum_{i=1}^{m} \sum_{x_j \in C_i} \|x_j - \bar{x}_i\|^2 \,, \tag{1}$$

and the goal is to find the optimal partition C^* such that

$$C^* = \arg\min_{C} E(C) \,. \tag{2}$$

The application of interest in this work suggests that the exemplars will be d-dimensional vectors encoding information about the original image, a transcoding operation, and the resulting file size and quality; the latter two being the quantities to predict.

The original compressed image I_j is represented by a tuple (QF_j, w_j, h_j, f_j), where QF_j is the original quality factor with which the image I_j was compressed (and in this work we assume that the quality factor complies with the semantics defined by the Independent JPEG Group [13], that is, it varies from 0 to 100 by increments of 1), w_j and h_j its width and height respectively, and f_j is the compressed file size of image I_j. The transcoding parameters form a tuple (QF_{out}, z) describing the new quality factor $QF_{out} \in \{10, 20, \ldots, 100\}$ and the scaling $z \in \{0.1, 0.2, \ldots, 1.0\}$ to apply to the image (thereby yielding 100 different possible transcodings). Applying the transcoding parameters (QF_{out}, z) to the image I_j yields an image with resolution $zw_j \times zh_j$ compressed with a new quality factor QF_{out}, resulting in an observed quality of $q(I_j, QF_{out}, z)$, and a relative file size of $f(I_j, QF_{out}, z)$ expressed as a ratio of the observed transcoded file size to the original file size.

The resulting quality measured between the original and the transcoded image is assessed by a quality metric; in our experiments, we opted for SSIM which we deem to be more representative of perceived quality than the PSNR [14]. In the case where the resolution of transcoded image differs from that of the original image (that is, $z \neq 1$), the transcoded image is scaled back to the original resolution for quality comparison. In all cases, scaling is performed using a Blackman filter, chosen for its spectral characteristics [15]. In previous works, we used a more sophisticated approach based on viewing conditions that depended strongly on the receiving device; in this work, we will omit this consideration, for the sake of simplicity, and use the original resolution as the viewing condition.

To the original data from the image and the transcoding operation, we will add two *features* that allows *a priori* knowledge impossible to be learned from clustering alone to be exploited. The first feature is the average number of bits per pixel of image I_j, denoted b_j, which is a measure of the busyness, or complexity, of the image. The second feature is the difference in quality factors, $QF_{out} - QF_j$ which gives information about changes in file size and quality. We discuss their nature further in section 5.

The exemplars will be 9-dimensional vectors. For an image I_j, the associated exemplar is given by

$$x_j = (QF_j, w_j, h_j, b_j, QF_{out}, z, QF_{out} - QF_j, f, q) \ , \tag{3}$$

with transcoding parameters (QF_{out}, z), and where, by abuse of notation, f and q stand for $f(I_j, QF_{out}, z)$ and $q(I_j, QF_{out}, z)$, respectively.

Solving eq. (2) exactly is an NP-Hard problem [16–18], and therefore we have to resort to approximate algorithms, such as K-Means [19], an algorithm very similar to LBG [20], which, while sub-optimal, has been shown to converge rapidly under most circumstances [21].

The K-Means algorithm proceeds iteratively until satisfactory convergence is reached. The initialization consists of randomly choosing m vectors from X (without replacement) to be the initial prototypes, the $\bar{X}_0$. Iterations proceed as follows. At iteration $t = 1, 2, \ldots$, for each vector $x_j \in X$, we find the closest prototype $\bar{x}_{t-1,i} \in \bar{X}_{t-1}$ such that

$$i = \arg\min_i \|x_j - \bar{x}_{t-1,i}\|^2$$

and assign the vector x_j to the subset $C_{t,i}$. All vectors assigned to a same subset $C_{t,i}$ are then averaged (by definition of centroids using Euclidean distance) to yield $\bar{x}_{t,i}$, a prototype for the next iteration. The algorithm iterates until satisfactory convergence is reached, that is, the error $E(C_t)$, as given by eq. (1), fails to diminish significantly from $E(C_{t-1})$. In our experiments, this threshold was set to a relative error reduction of $\alpha = 10^{-6}$ or less. Algorithm 1 details the procedure as pseudo-code.

The L_2 norm considered for eq. (2) and the lack of a distance matrix in the prob-

Algorithm 1 K-Means pseudo-code.

```
{Inputs:
   X, the exemplars
   $m$, the number of prototypes,
   $\alpha$, the convergence threshold}
$t \leftarrow 0$
$E_0 \leftarrow$ BIG_NUM
$\bar{X}_0 \leftarrow m$ random vectors from $X$ (without replacement)
for all $x_j \in X$ do
   $a_j \leftarrow \arg\min_i \|x_j - \bar{x}_{0,i}\|^2$ {compute membership}
end for
repeat
   $t \leftarrow t + 1$
   $n_t \leftarrow \{0, 0, \ldots, 0\}$ {$m$ elements, the $n_{t,i}$}
   $\bar{X}_t \leftarrow \{0, 0, \ldots, 0\}$ {$m$ elements, the $\bar{x}_{t,i}$}
   for all $x_j \in X$ do
      $i \leftarrow a_j$ {reuse previously computed membership}
      $\bar{x}_{t,i} \leftarrow \bar{x}_{t,i} + x_j$ {update centroid}
      $n_{t,i} \leftarrow n_{t,i} + 1$ {update the number of exemplars in this subset}
   end for
   for $i = 1$ to $m$ do
      $\bar{x}_{t,i} \leftarrow \bar{x}_{t,i}/n_{t,i}$ {normalize centroids}
   end for
   $E_t = 0$
   for all $x_j \in X$ do
      $i \leftarrow a_j \leftarrow \arg\min_i \|x_j - \bar{x}_{t,i}\|^2$ {updating membership}
      $E_t \leftarrow E_t + \|x_j - \bar{x}_{t,i}\|^2$
   end for
until converges($E_{t-1}, E_t, \alpha$)
{Outputs:
   $\bar{X}_t$, the prototypes}
```

lem formulation imply that the exemplars must lie in an isotropic space; in other words, all dimensions are spread along comparable scales. If the exemplars do not lie in such a space, the usual approach is to use principal component analysis, or similar techniques, to project the exemplars onto a vector space that provides isotropism [22]. Results suggest that, in our case, simple dimension-wise normalization by the dimension average suffices to provide satisfactory results.

What remains is to decide on the number of clusters, m. The number of clusters m is subject to a variety of trade-offs between prediction accuracy and computational cost of optimization and prediction, issues discussed further in section 5, but we will show that m does not need to be exceedingly large to outperform previously proposed predictors [9–11].

3 Prediction Algorithms

The data set used in the experiments for training and testing is composed of approximately 73000 JPEG images obtained by a Web crawler with highly popular Web sites as origination points [9]. For the experiments presented in this paper, the images have been split 90/10 for disjoint training and test sets, and each image was subjected to 100 different transcodings (corresponding to all possible quality factors $QF_{out} \in \{10, 20, \ldots, 100\}$ and scalings $z \in \{0.1, 0.2, \ldots, 1.0\}$), yielding approximately 6 570 000 training exemplars and 730 000 test exemplars, each modeled after eq. (3).

The first predictor we proposed [9], denoted here JQSP1, for JPEG Quality and Size Predictor, quantizes QF_{in} (corresponding to the QF_j of this work), QF_{out}, and z to $\widetilde{QF}_{in}$, $\widetilde{QF}_{out}$, and $\widetilde{z}$, respectively, in order to use a fixed-density table for the predictions (the "tilde" notation denotes quantized values). In [9], we constrained $\widetilde{QF}_{in}$ and $\widetilde{QF}_{out}$ to the set $\{10, 20, \ldots, 100\}$, and $\widetilde{z}$ to $\{0.1, 0.2, \ldots, 1.0\}$; resulting in the indices of a $10 \times 10 \times 10$ table whose entries contained the predictions. The predictions, of both resulting file size and resulting quality, are computed as the centroids of the exemplars with corresponding $\widetilde{QF}_{in}$, $\widetilde{QF}_{out}$, and $\widetilde{z}$.

The predictor presented in [11], denoted here JQSP2, also uses a table but rather than using QF_{in}, QF_{out} and z as indices, it uses $\widetilde{QF}_{in}$, $\widetilde{z}$, and a target file size to predict the transcoding parameter $\widehat{QF}_{out}$ and the resulting quality $\hat{q}$ (the "hat notation" denotes predicted values). The predictions are formulated as the centroid of the relative file size and the centroid of quality resulting from transcoding operations, selected amongst all training exemplars that minimize the difference with the target relative file size while maximizing quality. The details are given *in extenso* in [11], but suffice to say that the predictions are represented in a table indexed by $\widetilde{z}$ and $\widetilde{f}$, the target file size. The density of the table can be adjusted by varying the densities of the values that $\widetilde{z}$ and $\widetilde{f}$ can take; in our experiments we constrained $\widetilde{z} \in \{0.1, 0.2, \ldots, 1.0\}$, and $\widetilde{f}$ varies from 0.001 to 1.0 by steps of 0.001, thus minimizing errors due to quantization on f.

The variant of K-Means presented in Algorithm 1 is not very sensitive to the initial conditions (that is, the randomly chosen vectors that form $\bar{X}_0$) but its stochastic nature nevertheless demands that we perform several independent optimizations. In our case we opted for 30, in order to obtain a good solution by selecting the optimization with the smallest error as defined by eq. (1). The distribution of errors is presented in section 4 and discussed in section 5.

For all algorithms, training was performed on the first 90% of the exemplars. Tests were conducted by submitting the remaining 10% of the exemplars comprising all possible transcodings (c.f. eq. (3)) to the various prediction algorithms, comparing the observed resulting file size f and quality q to the predicted file size $\hat{f}$ and quality $\hat{q}$.

The error considered in the results (Figs. 1 and 2) is the average absolute error. Because the quantities predicted lie in $[0,1]$, using a mean squared error would result in exceedingly small quantities (as, for example, $0.1^2 = 0.01$) which would exaggerate the methods' performance.

4 Results

The predictors JQSP1, JQSP2, and the proposed EJQSP are compared using the same test exemplars—10% of all exemplars, disjoint from the training set, as discussed in section 3. For each test exemplar given by eq. (3), the predictors were asked to predict f and q, and the absolute errors $f-\hat{f}$ and $q-\hat{q}$ were measured. The EJQSP predictor was optimized for arbitrary, but likely to be chosen, numbers of prototypes (namely, 10, 25, 50, 100, 250, 500, 1000, 2000, 3000, 4000, 5000, 10000, and 20000) to show how the performance of the proposed method increases with the number of prototypes.

Examining Fig. 1, we see that the EJQSP predictor breaks even, on the accuracy of prediction of the resulting file size, with the predictor JQSP1 using only 500 prototypes and with JQSP2 using 2000. The performance difference continues to increase with the number of prototypes, yielding an error $\approx 40\%$ smaller than JQSP1, and approximately 27% less than JQSP2 with 20 000 prototypes. While the EJQSP prediction error reduction from using 20 000 prototypes rather than 10 000 is $\approx 10\%$, it is obtained at the cost of doubling the run-time since, as we discuss further in section 5, finding the nearest neighbor is a linear time problem. Fig. 2, shows similar behavior for quality prediction. The EJQSP predictor breaks even with JQSP2 using approximately 1500 prototypes, but breaks even with JQSP1 at 4000; however, EJQSP ultimately yields an error that is $\approx 20\%$ less than that of JQSP2 and $\approx 12\%$ less than JQSP1. Both figures clearly demonstrate that the EJQSP prediction

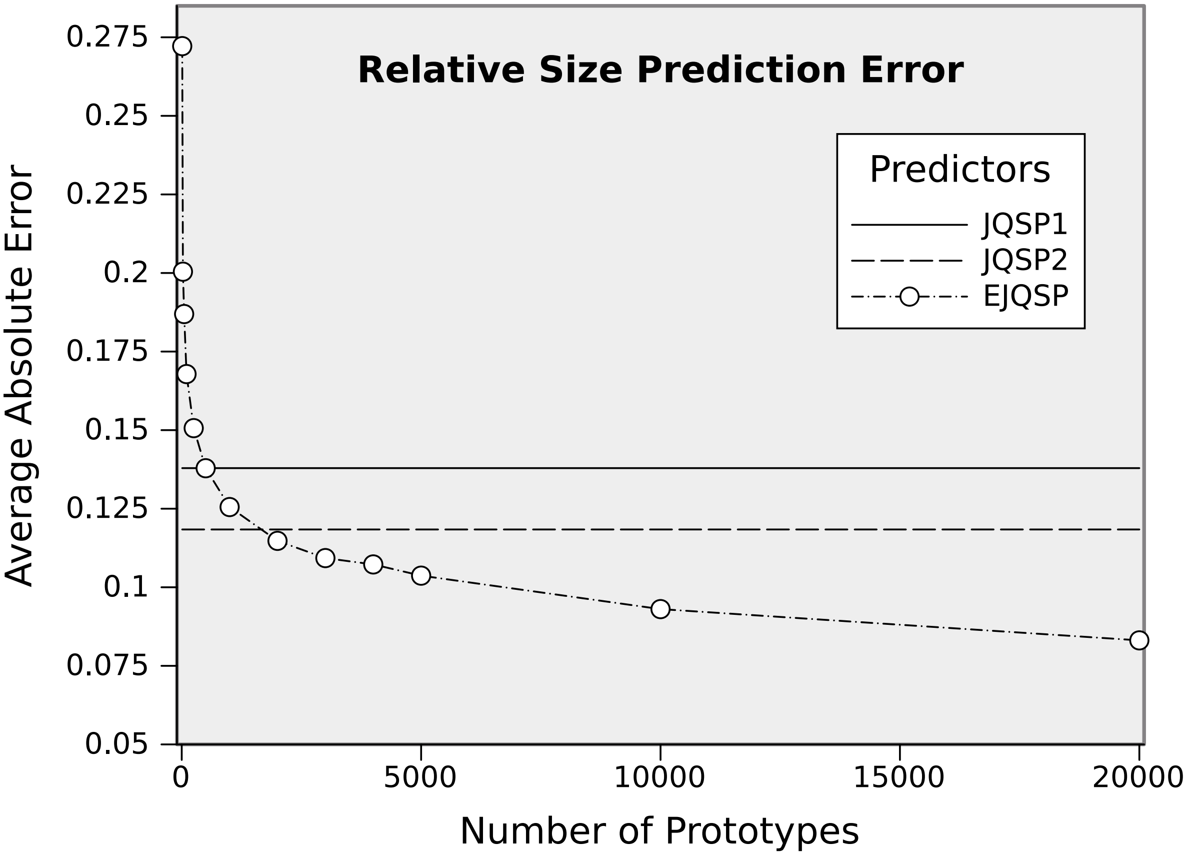

Figure 1: Average absolute error for relative size prediction.

accuracy in both file size and quality, increases with the number of prototypes; the large number of training exemplars enabling the use of a model with a large capacity [12].

Fig. 3 shows the distributions of the prediction error $\hat{f}-f$ (the error of the predicted relative file size $\hat{f}$ against the observed relative file size f) for the predictors considered in this study. Examining Fig. 3, we can see that the JQSP1 predictor, despite peaking near zero, exhibits strong skewness resulting in a definite propensity to overshoot the file size prediction. An algorithm using the JQSP1 predictor will therefore tend to take conservative decisions, probably failing to use the full file size budget. Predictor JQSP2, which also overshoots slightly, is already much better than JQSP1; not only does it not overshoot as much, it exhibits a stronger peak near zero, indicating that a much greater proportion of predictions are within an error of a few percent. Finally, we see that the novel EJQSP, using 20000 prototypes, has smaller overall relative file size prediction error, shows a general behavior comparable to JQSP2; however the distribution of errors for EJQSP

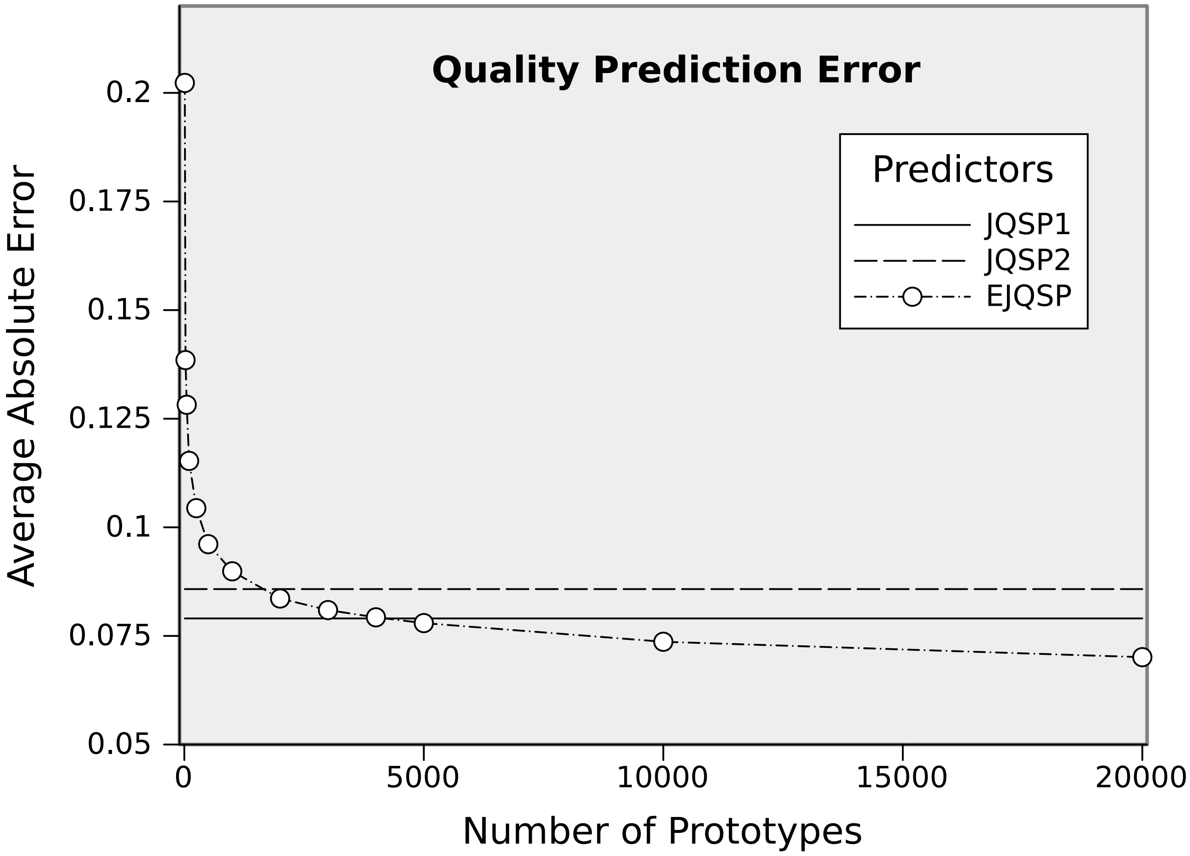

Figure 2: Average absolute error for SSIM prediction.

sports an even stronger peak around zero and it is less likely to overshoot.

While Fig. 3 shows that the individual relative file size predictors behave differently, Fig. 4 shows that for the resulting quality prediction error $\hat{q}-q$ (the error of the predicted resulting quality $\hat{q}$ versus the observed resulting quality q), the behavior is similar. For each predictor, the prediction error exhibits two distinct components: a component similar to a skewed Gaussian [23] and a strong peak at zero. While the JQSP2 predictor formulated better relative file size predictions than the JQSP1 predictor, the reverse is true for quality prediction, a fact that is also reflected in Fig. 2 where the overall error on quality prediction of JQSP1 is $\approx 8\%$ smaller than the error of JQSP2. While JQSP2 was designed to avoid overshooting on size prediction, there is no such provision for its quality prediction and, in fact, it overestimates resulting quality. The EJQSP predictor, like the JQSP1 predictor, tend to undershoot for prediction, however, the central peak and the more compact distribution yields much better overall results than previous predictors, as shown in all

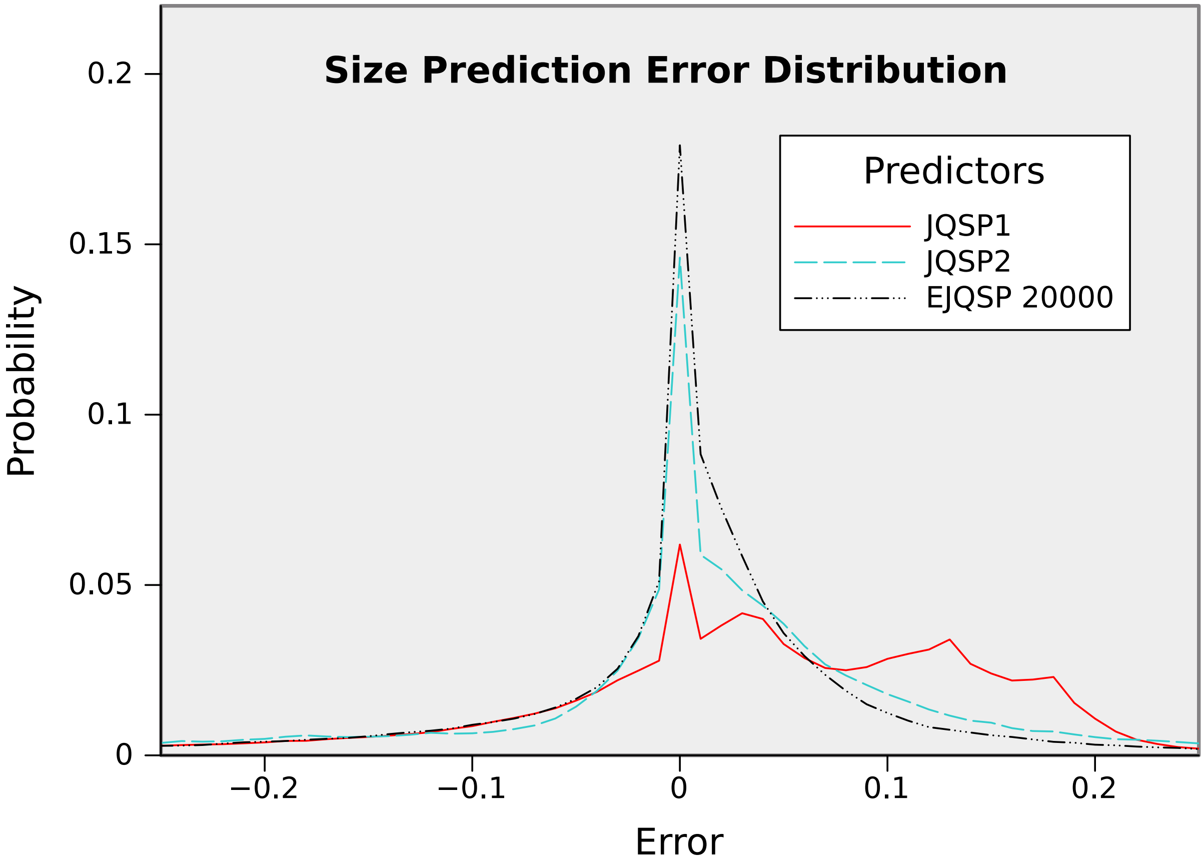

Figure 3: Distribution of errors on size prediction.

the figures.

5 Discussion

For all algorithms, training and test exemplars must be obtained. In our experiments, it meant subjecting every image from the database to 100 different transcodings—varying over $\widetilde{QF}_{out}$ and $\widetilde{z}$—to yield a sufficiently dense pool of transcoding examples. This process is of course very expensive but can be performed off-line as trends in image characteristics vary over the years at application-specific intervals.

Training the JQSP1 predictor is $O(n)$ in the number of exemplars, n. Training consists, for each exemplar $x_j \in X$, in quantizing the corresponding $\widetilde{QF}_{in}$, $\widetilde{QF}_{out}$, and $\widetilde{z}$, to index the

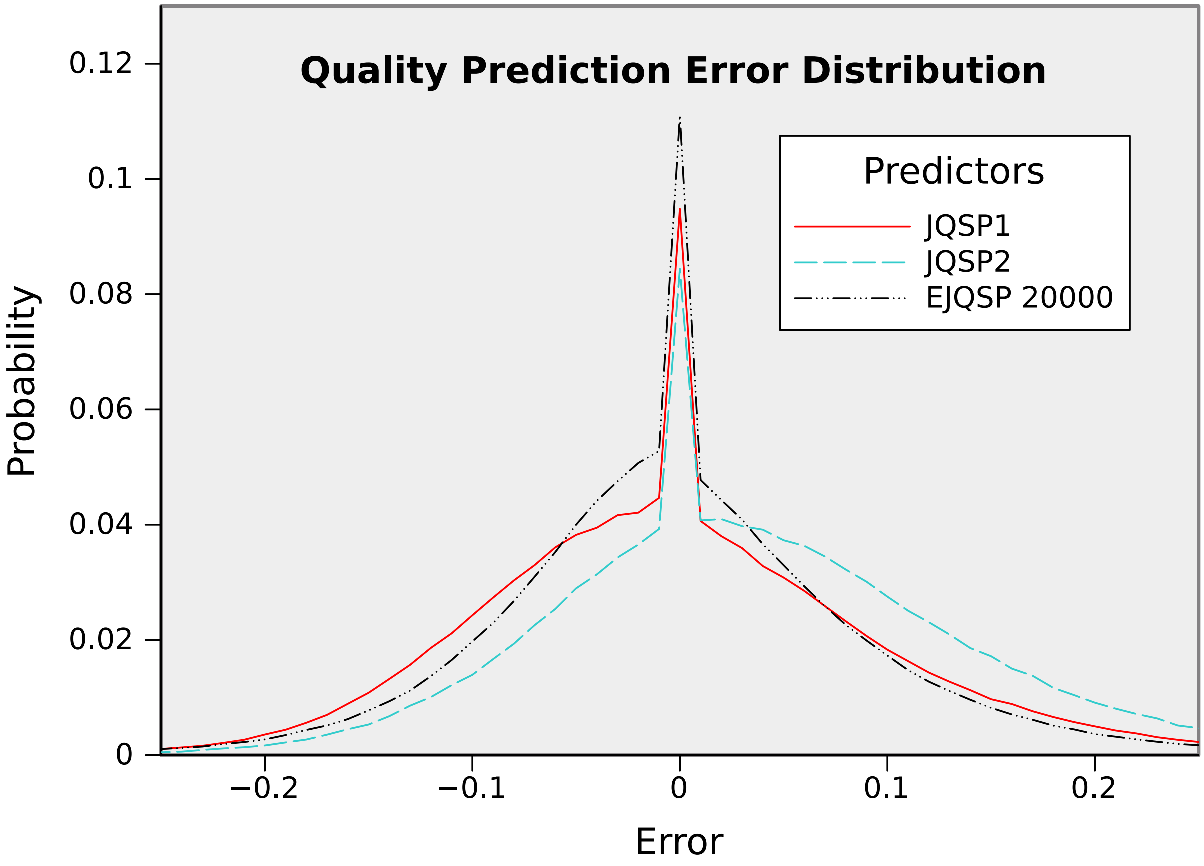

Figure 4: Distribution of errors on SSIM prediction.

table and accumulate the partial sums to compute the centroids. The centroid normalization is $O(|\widetilde{QF}_{in}|\,|\widetilde{QF}_{out}|\,|\widetilde{z}|)$, where, by abuse of notation, $|\widetilde{QF}_{in}|$, $|\widetilde{QF}_{out}|$, and $|\widetilde{z}|$ denote the number of values $\widetilde{QF}_{in}$, $\widetilde{QF}_{out}$, and $\widetilde{z}$ can take. The cost of normalization, proportional to the number of entries in the table, is negligible compared to n, since the number of entries must be very small compared to n in order to avoid the problem of context dilution [12].

The operating principle of the JQSP2 predictor is quite different, as it formulates a prediction on the QF_{out} needed to meet the target file size given a constraint z. It also predicts the resulting file size and quality and its design makes it unlikely to overshoot significantly on predicted file size [11], a fact confirmed by Fig. 3. The JQSP2 training phase constructs the table by going through all the images from the training set and selecting transcodings that are the closest to the desired file size with a scaling z without exceeding it to form the centroids. This results in a procedure $O(n\,|\widetilde{QF}_{in}|\,|\widetilde{z}|\,|\widetilde{f}|)$, where, again, by abuse of notation, $|\widetilde{f}|$ denotes the number of values $\widetilde{f}$ can take.

EJQSP formulates its prediction using clustering, as explained in section 2. Exact clustering, that is, solving eq. (2) for EJQSP exactly is NP-Hard [16–18], as we observed earlier, justifying the use of an approximate algorithm such as K-means. An iteration of K-Means is $O(m\,n\,d)$ for m prototypes and n exemplars in $\mathbb{R}^d$. Since K-means converges extremely rapidly, $O(\lg n)$ iterations seem to be sufficient for moderately large m and n; leaving an optimization complexity of $O(d\,m\,n\lg n)$.

Predicting a value using algorithms JQSP1 and JQSP2 is an $O(1)$ process, since it suffice to quantize the query parameters and use them directly to index the table. In both case, the $O(1)$ figure assumes uniform quantization on all parameters; other quantization schemes will lead to different complexities and prediction performance; however, in this paper we reproduced the testing condition of previous works [9–11]. The predictor we propose here formulates its prediction in $O(m\,d)$, where m is the number of prototypes in $\mathbb{R}^d$, as computing the exact nearest neighbor requires linear time, although it can be made sub-linear with preprocessing [18]; and approximate solutions [24] may not be welcomed as they are likely to introduce errors.

Adding features that exploit *a priori* knowledge usually helps classification and prediction tasks. Specifically, the bits-per-pixel feature will help distinguish images of comparable resolution but with quite different characteristics. For example, an image of a featureless blue sky will behave differently under transformation than a busy, detailed image, and the predictor must be able to take this information into account. The other feature, $QF_j - QF_{out}$, helps because it groups together similar drops in quality onto similar hyperplanes, a fact we verified experimentally.

JQSP1 uses a $10\times10\times10$ table with two values per entry (relative file size and quality), and while the table density can be adjusted, it is unlikely that it poses any kind of problems in a server-type environment as the table can be made a read-only singleton and that look-up is $O(1)$. Furthermore, the semantics of the quality factor imposes an upper bound on table density of $O(|\widetilde{z}|)$ with a hidden constant of 100×100 which corresponds to the density of QF_j and QF_{out}, where, again by abuse of notation, $|\widetilde{z}|$ denotes the number of values $\widetilde{z}$ can take. The same applies for JQSP2, but instead of a term in $|\widetilde{z}|$ we will have an additional term in $|\widetilde{f}|$, the number of values $\widetilde{f}$ can take. For EJQSP, the storage needed for the clusters is $O(m\,d)$, where m is the number of prototypes in $\mathbb{R}^d$.

There are trade-offs between storage and complexity to consider; one can adjust the parameters of the various algorithms to obtain similar storage consumptions, but storage, especially for server-side applications, is likely to be negligible. The algorithmic complexity of querying the predictors is moderate in all cases; leaving prediction accuracy as the foremost factor to consider. We have shown, from the results in section 4, that the proposed EJQSP predictor outperforms the previously proposed JQSP1 and JQSP2 predictors significantly, both for resulting file size and quality.

6 Conclusion

Despite formulating predictions in linear time in the number of prototypes (when considering the dimensionality of the vector space as constant), rather than the constant time JQSP1 and JQSP2 predictors, the proposed solution in this work, the EJQSP predictor, using 20 000 centroids, yields significantly better prediction of resulting file size and quality. It also yields ≈ 40% smaller prediction error on file sizes and ≈ 12% on quality compared to JQSP1, while yielding ≈ 27% smaller on size and ≈ 20% on quality compared to JQSP2. The new predictor, with its reduced error, can be combined with systems such as those presented in [10,11] to yield even more efficient transcoding systems, whether for universal media access or more specific applications such as multimedia messaging.

Acknowledgements

This work was funded by Vantrix Corporation and by the Natural Sciences and Engineering Research Council of Canada under the Collaborative Research and Development Program (NSERC-CRD 326637-05). Authors can be contacted at `pigeon@iro.umontreal.ca` and `stephane.coulombe@etsmtl.ca`

References

[1] R. Mohan, J. R. Smith, and C.-S. Li, "Adapting Multimedia Internet Content for Universal Access," *IEEE Trans. Multimedia*, vol. 1, no. 1, pp. 104–114, Mar. 1999.

[2] R. Han, P. Bhagwat, R. LaMaire, T. Mummert, V. Perret, and J. Rubas, "Dynamic Adaptation in an Image Transcoding Proxy for Mobile Web Browsing," *IEEE Personal Communications Magazine*, vol. 5, no. 6, pp. 8–17, 1998.

[3] S. Coulombe and G. Grassel, "Multimedia Adaptation for the Multimedia Messaging Service," *IEEE Communication Magazine*, vol. 42, no. 7, pp. 120–126, July 2004.

[4] Open Mobile Alliance, "Enabler Test Specification (for Conformance) for MMS Candidate Version 1.3," Oct. 2010.

[5] J. Ridge, "Efficient Transform-Domain Size and Resolution Reduction of Images," *Signal Processing: Image Communication*, vol. 18, no. 8, pp. 621–639, Sept. 2003.

[6] V. Ratnakar and V. Ivashin, "File Size Bounded JPEG Transcoder," May 2001, US Patent 6,233,359.

[7] Z. Lei and N. Georganas, "Accurate bit allocation and rate control for DCT domain video transcoding," in *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2002)*, 2002, vol. 2, pp. 968–973.

[8] J. R. Smith, R. Mohan, and C.-S. Li, "Content-Based Transcoding of Images in the Internet," *Int. Conference on Image Processing (ICIP)*, 1998.

[9] S. Pigeon and S. Coulombe, "Computationally Efficient Algorithms for Predicting the File Size of JPEG Images Subject to Changes of Quality Factor and Scaling," in *Procs. 24th Queen's University Biennial Symposium on Communications*, 2008.

[10] S. Coulombe and S. Pigeon, "Quality-Aware Selection of Quality Factor and Scaling Parameters in JPEG Image Transcoding," in *Procs. IEEE 2009 Computational Intelligence for Multimedia, Signal, and Video Processing (CIMSVP)*.

[11] S. Coulombe and S. Pigeon, "Low-Complexity Transcoding of JPEG Images with Near-Optimal Quality Using a Predictive Quality Factor and Scaling Parameters," *IEEE Trans. Image Processing*, vol. 19, no. 3, pp. 712–721, Mar. 2010.

[12] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer, 2009.

[13] "The Independent JPEG Group," http://www.ijg.org/.

[14] Z. Wang, A. C. Bovick, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Trans. Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[15] R. B. Blackman and J. Tukey, *The Measurement of Power Spectra, from the Point of View of Communications Engineering*, Dover, 1959.

[16] D. Aloise, A. Deshpande, P. Hansen, and P. Popat, "NP-Hardness of Euclidean Sum-of-Squares Clustering," *Machine Learning*, vol. 75, no. 2, pp. 245–248, Apr. 2010.

[17] A. Vattani, "K-Means Require Exponentially Many Iterations Even in the Plane," in *Procs. 25th Symposium on Computational Geometry*, 2009.

[18] M. Mahajan, P. Nimbhorkar, and K. Varadarajan, "The Planar K-Means Problem is NP-Hard," in *WALCOM: Algorithms and Computation*, 2009, pp. 274–285, (LNCS #5431).

[19] S. P. Lloyd, "Least Squares Quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

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

[21] 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.

[22] I. T. Jolliffe, *Principal Component Analysis*, Springer, 2002.

[23] A. Azzalini, "A Class of Distributions that Includes the Normal Ones," *Scand. J. Stat.*, vol. 12, pp. 171–178, 1985.

[24] P. Indyk and R. Motwani, "Approximate Nearest Neighbor: Towards Removing the Curse of Dimensionality," in *Procs. 30th Annual ACM Symposium on Theory of Computing (STOC'98)*, 1998, pp. 604–613.