COMPRESSION COLOR SPACE ESTIMATION OF JPEG IMAGES USING LATTICE BASIS REDUCTION

Ramesh Neelamani, Richard G. Baraniuk *

Department of Electrical and Computer Engineering
Rice University
Houston, TX 77005

Ricardo de Queiroz

800 Phillips Road
Xerox Corporation
Webster, NY 14580

ABSTRACT

Given a color image that was quantized in some hidden color space (termed compression color space) during previous JPEG compression, we aim to estimate this unknown compression color space from the image. This knowledge is potentially useful for color image enhancement and JPEG re-compression.

JPEG quantizes the discrete cosine transform (DCT) coefficients of each color plane independently during compression. Consequently, the DCT coefficients of such a color image conform to a lattice. We exploit this special geometry using the lattice reduction algorithm used in cryptography to estimate the compression color space. Simulations verify that the proposed algorithm yields accurate compression space estimates.

1. MOTIVATION

Color images can be expressed in many equivalent representations or color spaces [1]. Each color space requires three independent values to describe a color, eg., the *RGB* color space expresses each color in terms of its red, green and blue components. Most color spaces are related to each other by linear transformations that is captured by a 3×3 matrix.

JPEG is a commonly used standard to compress still color images [2] (see Figure 1). However, the choice of the color space in which quantization is performed during JPEG compression (henceforth termed as the compression color space) is not standardized; this choice can vary in different JPEG implementations. The knowledge of the compression color space used during previous JPEG compression is often lost in its current uncompressed representation. For example, a display or a printing driver is just handed the bitmap of the uncompressed image with no information about the compression color space. To enhance or re-compress such color images, knowledge of the compression color space would be useful[3].

In this paper, we address the following problem: given a color image that is currently represented in some arbitrary color space ABC, but was quantized in some unknown color space PQR during previous JPEG compression, estimate the linear transformation relating the PQR color space to the ABC color space.

2. PROBLEM GEOMETRY

The coefficients of an image subjected previously to JPEG compression conform to a regular geometric structure, which can be exploited to estimate the compression color space. The inherent geometry can be understood by analyzing the operations that a previously JPEG-compressed image is subjected to.

* Supported by the NSF grant CCR–99–73188, DARPA/AFOSR grant F49620–97–1–0513, and ONR grant N00014–99–1–0219.

2.1. JPEG compression, decompression, and transformation

Consider a color image that is currently represented in the ABC color space (see Figure 1). Assume that PQR is the compression color space used by JPEG (see Figure 1). The compression color space PQR is assumed to be unknown. Such a color image undergoes the following operations to reach its current representation in the ABC color space.

- **JPEG compression:** JPEG performs the following operations on each color plane P, Q, and R independently:
  1. Take the DCT of each 8×8 block in the chosen plane.
  2. Let i denote one of the 64 resulting DCT coefficients, and q_i denote the corresponding quantization step size. Quantize the i^{th} DCT coefficient of each 8×8 block from Step 1 to the closest integer multiple of q_i. Let c_i denote the i^{th} DCT coefficient of one such block. Then, c_i is quantized to $N_i q_i$, where $(c_i/q_i) - 0.5 \leq N_i < (c_i/q_i) + 0.5, N_i \in \mathbb{Z}$.

  The compressed image is stored by retaining the quantized DCT coefficients of each color plane. Sometimes sub-sampling is also employed after Step 2 to achieve further compression [2]. However, in this paper, we have assumed that sub-sampling is not performed.
- **JPEG decompression:**
  1. Take the inverse DCT of the 8×8 blocks of quantized coefficients.
  2. Round-off resulting pixel values to the nearest integer so that they lie in the 0–255 range.
- **Color transformation:** To be represented in the current ABC representation, the image would undergo a transformation (assumed to be linear) from PQR to ABC space.

2.2. Ideal geometry of previously JPEG-compressed image

Consider an arbitrary 8×8 color image block that the DCT acts on during JPEG compression in the PQR space. Let c_i^P, c_i^Q, and c_i^R denote the respective i^{th} frequency DCT coefficients of the P, Q, and R planes in the chosen 8×8 color image block. JPEG quantizes the DCT coefficients of the each plane *independently* to $N_i^P q_i^P$, $N_i^Q q_i^Q$, and $N_i^R q_i^R$, where the notations follow from Step 2 in JPEG compression described in Section 2.1. All the i^{th} DCT frequency coefficients from the different 8×8 blocks in the image are subjected to the same quantization step size (q_i^P, q_i^Q, and q_i^R for the P, Q, and R planes respectively). Consider the 3-dimensional (3-d) vector of quantized DCT coefficients $[N_i^P q_i^P \;\; N_i^Q q_i^Q \;\; N_i^R q_i^R]^T$. Due to independent quantization of each plane, all such 3-d vectors of i^{th} frequency DCT coefficients lie on a rectangular box grid with edge-lengths equal to the quantization step sizes (see Figure 2).

If this compressed image is now represented in some other color space ABC using some linear transformation from PQR (see Figure 2), then the corresponding 3-d vectors of DCT coefficients in the ABC space do not lie on a rectangular box grid, but on a *parallelepiped*[1] grid, assuming that no round-off is performed during JPEG decompression in the PQR space. The edges of the parallelepiped are determined by the column vectors of the 3×3 color transform from PQR to ABC, which we henceforth denote by T. The typical color space in which the 3-d vectors of DCT coefficients lie on a rectangular box grid for *each* DCT frequency is the PQR color space. Thus, the geometry of the DCT coefficients can be exploited to determine the compression color space PQR from the image represented in the ABC color space.

2.3. Round-off errors perturb coefficient geometry

Round-offs employed during JPEG decompression (see Figure 1 and Step 2 in JPEG decompression in Section 2.1) perturb the DCT coefficient values. Consequently, the 3-d vectors of DCT coefficients in the PQR color space representation lie only approximately on the rectangular-box grid (see Figure 2). Let $\vec{E}$ denote the 3-d error vector between the vector of DCT coefficients before and after round-off. Then, from [3], the perturbations in the 3-d DCT coefficient vectors in the PQR space can be statistically modeled by a truncated 3-d Gaussian

$$P(\vec{E}) \quad \propto \quad \exp(-6\|\vec{E}\|^2), \text{ where } \vec{E} \in [-S, S]^3,$$

where $P(E)$ denote the probability density function (PDF) of $\vec{E}$, and $[-S, S]^3$ denotes the cube centered at the origin with edge-length $2S$ that supports of the truncated Gaussian. S changes with the different DCT frequencies; the maximum value for S is 4 [3].

After transformation from the PQR space to ABC space, the 3-d perturbation error vector $\vec{E_{ABC}}$ in the ABC space is given by $\vec{E_{ABC}} = T\vec{E}$. Hence,

$$P(\vec{E}_{ABC}) \quad \propto \quad \exp(-6\|T^{-1}\vec{E}_{ABC}\|^2), \;\; \vec{E}_{ABC} \in T[-S, S]^3,$$

where $T[-S, S]^3$ denotes the cube $[-S, S]^3$ transformed by the color transform T. The exact PDF is dependent on the unknown transformation T, which is inconvenient. We approximate the PDF for the perturbation error vector $\vec{E_{ABC}}$ in the ABC space as a truncated Gaussian with increased support as

$$\begin{aligned} P(\vec{E}_{ABC}) \quad &\propto \quad \exp(-6\|\vec{E}_{ABC}\|^2), \\ &\qquad \vec{E}_{ABC} \in [-5, 5]^3. \end{aligned} \tag{1}$$

Though this approximation is coarse, we will see that we still obtain satisfactory estimation results.

3. LATTICE REDUCTION ALGORITHM

Lattices are regular arrangements of points in space, whose study arises in both number theory and crystallography. Consider an ordered set of m vectors $b_1, b_2, \ldots b_m$. Then a *lattice* L spanned by these vectors consists of all *integral* linear combinations $\gamma_1 b_1 + \gamma_2 b_2 + \ldots + \gamma_m b_m, \gamma_i \in \mathbb{Z}$.

The structure in Figures 2(a) and (b) are both examples of 3-d lattices. Our need to exploit the lattice structure offered by the problem prompts us to invoke lattice reduction algorithms discovered in field of number theory. Given a set of vectors b_i's that lie on a lattice, the goal of lattice reduction is to find an ordered set of *basis* vectors for the lattice spanned by the b_i's such that [4]

1. the basis vectors are maximally orthogonal,
2. the shorter basis vectors appear first in the ordering.

A major breakthrough in this long-time open problem was the discovery of the LLL algorithm [5] to perform *lattice reduction* in polynomial time. LLL algorithms have since proved invaluable in many areas of mathematics and computer science, especially in algorithmic number theory and cryptology [6, 4]. Lattice reduction is achieved by using a sequence of very simple operations on the vectors b_i's. These operations are

1. Change the order of the basis vectors.
2. Add to one of the vectors b_i an integral multiple of another vector b_j. Note that the vectors resulting from such integral operations still lie on the same lattice.
3. Delete any resulting zero vectors.

4. LATTICE REDUCTION FOR COMPRESSION SPACE ESTIMATION

In the absence of round-off errors, 3-d vectors of ABC color space DCT coefficients would exactly lie on a lattice. Hence the LLL algorithm applied to these 3-d vectors would provide a set of almost orthogonal basis vectors that spans the parallelepiped lattice in Figure 2 (b). However, 3-d vectors of DCT coefficients are perturbed from the exact lattice locations due to round-off. Since the perturbation errors in the DCT coefficient vectors get amplified during the arithmetic operations used by the LLL algorithm, a direct implementation of the LLL algorithm is not feasible to estimate the desired basis vectors that span the approximate parallelepiped. Fortunately, since there are many 8×8 blocks in the image, we often have many realizations of 3-d DCT vectors that belong to the same parallelepiped lattice location. This provides us with an opportunity to mitigate the noise in the 3-d DCT vectors, thereby resulting in a more robust lattice estimation algorithm.

We propose the following lattice estimation algorithm to fuse our knowledge about the statistics of the round-off noise with the LLL algorithm. The steps in the algorithm are as follows:

1. Choose a DCT frequency. Take the 3-d histogram of the 3-d DCT coefficient vectors from the different 8×8 blocks.
2. Sort the locations of the histogram bins in descending order of the histogram values obtained in Step 1. This ensures that the LLL algorithm is initiated with the largest coefficient, which should be the least noisy vector.
3. Choose the first location vector on the sorted list that lies outside the cube $[-5, 5]^3$ as a basis vector to the lattice. Any vector within the cube $[-5, 5]^3$ could potentially be a noisy realization of origin $[0\ 0\ 0]^T$, and is hence ignored.
4. Choose the next location vector. If there are no more vectors left in the list, then exit.
5. Calculate the error vector between the currently chosen vector and the closest vector that lies on the lattice spanned by the current set of basis vectors. The calculation of the error vector invokes a slight variant of the LLL algorithm.
6. If the error vector calculated in Step 5 lies outside the cube $[-5, 5]^3$, then the currently chosen vector does not lie in the span of the current set of basis vectors. Hence add the currently chosen vector to list of basis vectors. Perform LLL on this set of basis vectors. Go to Step 4.
7. If the error vector lies inside the cube $[-5, 5]^3$, then the currently chosen vector lies in the span of the current set of basis vectors. Add the current vector to the list of vectors that already lie in the span of the current basis, and massage the basis vectors to minimize the cumulative probability of error (see Appendix A for details). Go to Step 4.

[1] A solid with six faces, each of which is a parallelogram

For each DCT frequency, the above algorithm yields a set of basis vectors for lattice that the 3-d vector of the respective DCT coefficients approximately lie on.

Since any set of lattice basis vectors is not unique, the final piece in the puzzle is the deduction of the color transform from the estimated lattice basis vectors from the different DCT frequencies. Let L_i be the estimated set of lattice basis vectors for DCT frequency i. Then $L_i = T[\Lambda_i]U_i$, where $[\Lambda_i]$ is a diagonal matrix with entries given by the respective quantization step sizes used during compression in the PQR space. U_i is a unit-determinant matrix with integer entries to account for the non-uniqueness of lattice basis vectors. An estimate of any scaled version of the color transform matrix T such as $T[\Lambda_i]$ would solve our problem, since the color transform matrix is assumed to have unit-norm column vectors. Hence we need to estimate and undo the effect of U_i from L_i to obtain the color transform estimate. Let $L_j = T[\Lambda_j]U_j$ be estimated set of lattice basis vectors for DCT frequency j. We observe that $(L_i U_i^{-1})^{-1}(L_j U_j^{-1})$ is a diagonal matrix. U_i^{-1} and U_j^{-1} are also integer matrices; hence we can undo the effects of U_i and U_j from L_i and L_j respectively by trying different integer addition and subtraction operations on the columns of L_i and L_j, so that $(L_i U_i^{-1})^{-1}(L_j U_j^{-1})$ is diagonal. With heuristics, this search can be performed very efficiently to obtain the desired color transform estimate.

5. RESULTS

To verify our proposed algorithm, we used a test color image that was JPEG-compressed in the *ITU.BT-601 YCbCr* space with quality factor 70. The *Cb* and *Cr* planes were not sub-sampled during compression. The uncompressed image was then transformed to the *RGB* space. Our algorithm operated in this *RGB* space, and tried to estimate the color transform matrix from *ITU.BT-601 YCbCr* to current *RGB*.

The actual transform T from *ITU.BT-601 YCbCr* to *RGB* with columns normalized to unity is

$$T = \begin{bmatrix} 0.5774 & 0.0005 & 0.8910 \\ 0.5774 & -0.1904 & -0.4540 \\ 0.5774 & 0.9817 & 0.0006 \end{bmatrix} \tag{2}$$

The lattices estimated by our proposed algorithm for the DCT frequencies $[2, 2]$ and $[2, 3]$ respectively were

$$\begin{bmatrix} 7.00 & -18.21 & -6.98 \\ 7.00 & 9.26 & -11.45 \\ 7.00 & 0.02 & 15.97 \end{bmatrix} \text{ and } \begin{bmatrix} 8.01 & -22.40 & -7.98 \\ 8.01 & 11.42 & -13.50 \\ 8.00 & 0.05 & 20.31 \end{bmatrix}.$$

Though the first two columns of the two matrices above are scaled versions of each other, the third column is not. This is easily fixed by adding the first columns to the respective third columns. The aligned lattice basis for the DCT frequencies $[2, 2]$ and $[2, 3]$ respectively become

$$\begin{bmatrix} 7.00 & -18.21 & 0.02 \\ 7.00 & 9.26 & -4.45 \\ 7.00 & 0.02 & 22.97 \end{bmatrix} \text{ and } \begin{bmatrix} 8.01 & -22.40 & -0.03 \\ 8.01 & 11.42 & -5.49 \\ 8.00 & 0.05 & 28.31 \end{bmatrix}.$$

The estimated color space $\widehat{T}$ obtained by normalizing the above matrices and averaging them is

$$\widehat{T} = \begin{bmatrix} 0.5775 & 0.0009 & 0.8911 \\ 0.5775 & -0.1903 & -0.4537 \\ 0.5771 & 0.9817 & -0.0015 \end{bmatrix}. \tag{3}$$

We can see that the estimated transform $\widehat{T}$ compares extremely well with the original color transform T. Columns 2 and 3 of $\widehat{T}$ have been interchanged, and the signs have been reversed to compare with T; however, the ordering and sign-changes are insignificant in practice.

6. CONCLUSIONS

In this paper, we estimate the unknown compression color space used during previous JPEG compression. This estimation is potentially important to enhance and re-compress such previously JPEG-compressed color images.

Our problem analysis shows that the image DCT coefficients of a previously JPEG-compressed image conform to an approximate lattice that can be exploited to determine the unknown compression color space. To estimate this geometry, we propose an estimation algorithm that fuses statistical noise reduction with the novel and powerful lattice reduction algorithm from number theory. The algorithm accurately estimates the desired compression color space during simulations.

We are currently working on improvements to the algorithm. Further, we are also working on incorporating the effects of subsampling, and non-linearities such as gamma correction into the estimation framework.

Appendix A: Updating the basis vectors

In this appendix, we update the estimated basis vectors by exploiting the multiple noisy realizations to mitigate the noise in the estimate. Let B_r denote the current set of lattice-reduced basis column vectors. Let D denote the matrix of 3-d DCT column vectors that have already been sorted through (see Step 7 in the proposed algorithm). Since all the vectors in D lie close to the lattice spanned by B_r, we can write $D = B_r\, S + \Delta$, where Δ is the matrix of the perturbation vectors, and S is a matrix with integral entries obtained using B_r before update. Typically, if then the B_r entries are close to the actual values, the S estimate is exact. Assuming each perturbation vector is independent of each other, and ignoring the finite support of the PDF in (1), we have

$$P(\Delta) = \exp(-6\|\Delta\|^2) \tag{4}$$

$$= \exp(-6\|D - B_r S\|^2), \tag{5}$$

where $\|.\|^2$ denotes the sum of squares of all entries in the matrix. The basis vectors are updated by differentiating the exponent $\|D - B_r S\|^2$ with respect to the B_r and setting it to zero. The updated basis vectors minimize the error probability in (4) assuming that the estimate of the integer matrix S is exact.

7. REFERENCES

[1] G. Sharma and H. Trussell, "Digital color imaging," *IEEE Trans. Image Processing*, vol. 6, no. 7, pp. 901–932, July 1997.

[2] W. Pennebaker and J. Mitchell, *JPEG, Still Image Data Compression Standard*, Van Nostrand Reinhold, 1993.

[3] Z. Fan and R. de Queiroz, "Maximum likelihood estimation of JPEG quantization table in the identification of JPEG compression history," in *Proc. IEEE Int. Conf. Image Processing — ICIP '00*, Vancouver, Canada, Sep. 2000.

[4] A. Joux and J. Stern, "Lattice reduction: A toolbox for the cryptanalyst," *Journal of Cryptology*, vol. 11, no. 3, pp. 161–185, 1998.

[5] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász, "Factoring polynomials with rational coefficients," *Mathematics Annalen*, vol. 261, pp. 515–534, 1982.

[6] P. Nguyen and J. Stern, "Lattice reduction in cryptology: An update," in *Lecture notes in Comp. Sci.* 2000, vol. 1838, pp. 85–112, Springer Verlag.

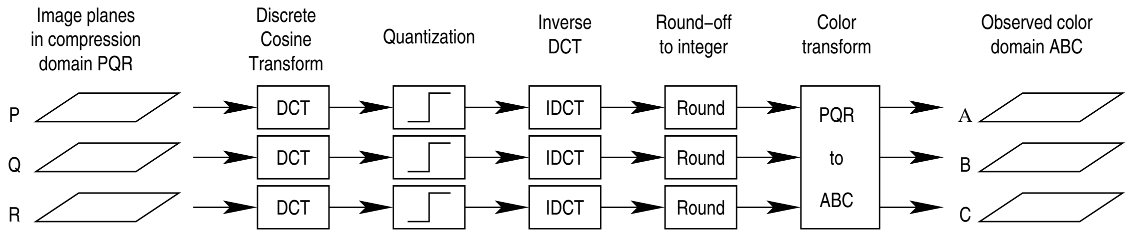

Fig. 1. *JPEG compression, decompression, and color transformation*: Assume that a JPEG implementation chooses some arbitrary color space PQR to perform compression. Then, JPEG operates independently on the three color planes P, Q, and R. During compression, JPEG first takes the discrete cosine transform (DCT) of 8×8 blocks in each plane, and second, quantizes each DCT coefficient to an integer multiple of some chosen quantization step size. The decompression algorithm first takes the inverse DCT, and second, rounds-off the pixel values to the nearest integer so that they lie between the conventional 0-255 range. Any decompressed image is often linearly transformed and represented in some arbitrary color space, say ABC. Often, the knowledge of the compression color space PQR is lost. In this paper, we seek to estimate the color transform from the compression color space PQR to the current color space ABC.

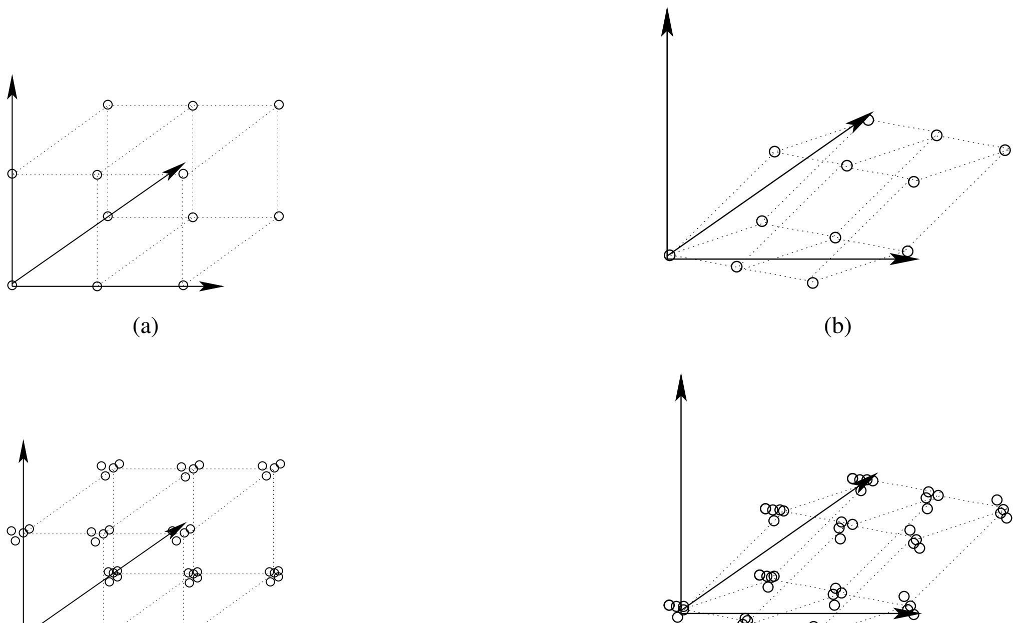

Fig. 2. *Lattice structures in the previously JPEG-compressed color image:* (a) DCT coefficient geometry in the compression color space PQR assuming the absence of round-off during JPEG decompression. All the 3-d vectors of DCT coefficients from the different 8×8 image blocks but same DCT frequency lie exactly on the vertices of a rectangular box. Each 3-d vector is denoted by a small circle in the figure. (b) DCT coefficient geometry in the observed color space ABC assuming round-off errors are absent. The 3-d vectors of DCT coefficients lies exactly on the vertices of a parallelepiped grid (formally termed as a lattice), whose edges are determined by the column vectors of the matrix transformation from the PQR to the ABC color space. Given these 3-d DCT coefficient vectors, the LLL algorithm [5] yields vectors that form the edges of the parallelepiped grid. (c) DCT coefficient geometry in the compression color space PQR assuming the presence of round-off during decompression. The 3-d vectors of DCT coefficients are slightly perturbed from the vertices of the rectangular-box grid. (d) DCT coefficient geometry in the observed color space ABC assuming round-off errors are present. The 3-d vectors of DCT coefficients are slightly perturbed from the vertices of the parallelepiped grid locations. Our proposed algorithm accurately estimates the vectors forming the edges of the parallelepiped grid from the perturbed 3-d DCT coefficient vectors.