Electronic Notes in Theoretical Computer Science 46 (2001)
URL: http://www.elsevier.nl/locate/entcs/volume46.html 10 pages

Lattice Algorithms for Compression Color Space Estimation in JPEG Images

Ramesh Neelamani [a,1,2], Ricardo de Queiroz [b,3], and Richard Baraniuk [a,1,4]

[a] *Department of Electrical and Computer Engineering*
Rice University
Houston, TX 77005

[b] *800 Phillips Road*
Xerox Corporation
Webster, NY 14580

Abstract

JPEG (Joint Photographic Experts Group) is an international standard to compress and store digital color images [5]. Given a color image that was previously JPEG-compressed in some hidden color space, 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 operates on the discrete cosine transform (DCT) coefficients of each color plane independently during compression. Consequently, the DCT coefficients of the color image conform to a lattice structure. We exploit this special geometry using the lattice reduction algorithm from number theory and cryptography to estimate the compression color space. Simulations verify that the proposed algorithm yields accurate compression color space estimates.

1 Motivation

Digital color images can be expressed in many possible equivalent representations or color spaces. Each pixel in a digital color image consists of three values, because the description of colors perceived by the human visual system requires three numerical components. Each three-component collection

[1] This work was supported by the National Science Foundation grant CCR–99–73188, DARPA/AFOSR grant F49620–97–1–0513, and ONR grant N00014–99–1–0219.

[2] Email: neelsh@rice.edu
[3] Email: queiroz@wrc.xerox.com
[4] Email: richb@rice.edu

such as {Red, Green, Blue} (RGB), and {Luminance Y, Chrominance Cr, Chrominance Cb} ($YCbCr$) is termed as a color space. Using vector notations, any color can be expressed as a 3-dimensional (3-d) vector with respect to a color space; the vector entries quantify the respective contributions of the color space components. Many color spaces are related to each other by linear transformations that are captured by 3×3 matrices. Hence a given color,and thereby any color image, can be represented in terms of another color space by transforming its 3-d vector representation using the 3×3 matrix. For an excellent tutorial on digital color imaging, refer to [6].

JPEG is a commonly used standard to compress still color images [5] (see Figure 1 for an over-view of the JPEG compression algorithm; further details are provided in Section 2). However, the choice of the color space used in the JPEG compression algorithm is not standardized; this choice can vary from one implementation of JPEG to other. [5] The knowledge of the compression color space used in a previously JPEG-compressed image 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; no information about the color space in which the image was previously compressed is given to it. To enhance or re-compress such color images, knowledge of the compression color space would be useful[1].

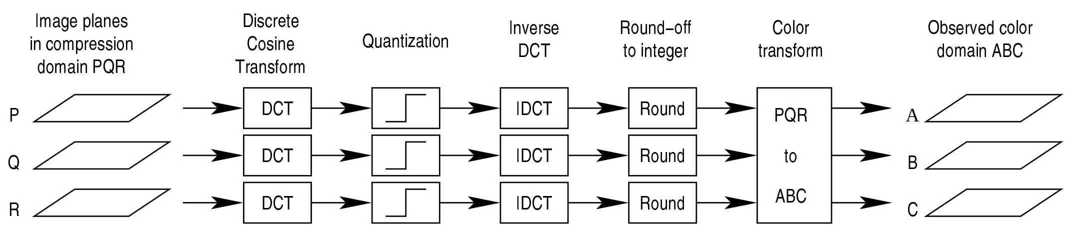

Fig. 1. *JPEG compression, decompression, and color transformation : Assume that a JPEG implementation chooses some arbitrary color space PQR to perform compression. Then, JPEG operates independently on the three color planes P, Q, and R. During compression, JPEG first takes the discrete cosine transform (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 using a 3×3 color transformation matrix, 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 observed color space ABC.*

In this paper, we address the following problem: given a color image that is currently represented in some arbitrary color space, say ABC, but was previ-

[5] Typically, $YCbCr$ space is chosen, because it facilitates better compression. However, this choice is not mandatory; in fact, many different flavors of $YCbCr$ such as *Kodak PhotoYCC* and *ITU.BT-601 YCbCr* exist, and are used in practice.

ously JPEG-compressed in some unknown color space, say PQR, estimate the linear transformation relating the PQR color space to the ABC color space.

2 Problem Geometry

The coefficients of an image previously subjected 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 undergone by a previously JPEG-compressed image.

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 the image was previously JPEG-compressed in the PQR color space. The knowledge of the compression color space PQR is assumed to be unknown or lost. The given color image would undergo the following operations to reach its current representation in its current ABC color space from the PQR color space.

- *JPEG compression:* During compression in the PQR space, JPEG essentially performs the following operations independently on each color plane P, Q, and R:
  (i) Split the color plane into 8×8 blocks. Take the DCT of each block in the chosen plane.
  (ii) 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 [5]. However, in this paper, we have assumed that sub-sampling is not performed.
- *JPEG decompression:* During decompression, the following operations are performed (see Figure 1):
  (i) Take the inverse DCT of the 8×8 blocks of quantized coefficients.
  (ii) 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. This transformation is characterized by a 3×3 matrix.

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 i denote one of the 64 possible frequencies in the DCT domain. 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 follows 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 $[N_i^P q_i^P \quad N_i^Q q_i^Q \quad N_i^R q_i^R]^T$. Due to independent quantization of each plane, all such 3-d vectors of i^{th} frequency DCT coefficients lie on a rectangular box grid, whose edges are determined by the quantization step size values.

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 no longer lie on a rectangular box grid. The 3-d DCT coefficient vectors lie on a *parallelepiped*[6] grid (see Figure 2), 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 any arbitrary color space ABC.

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. Due to round-off errors, 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 $\boldsymbol{E}$ denote the 3-d error vector between the vector of DCT coefficients before and after round-off. Then, from [1], the perturbations in the 3-d DCT coefficient vectors in the PQR space can be statistically modelled by a truncated 3-d Gaussian

$$P(\boldsymbol{E}) \propto \exp(-6\|\boldsymbol{E}\|^2), \text{ where } \boldsymbol{E} \in [-S, S]^3, \tag{1}$$

where $P(E)$ denote the probability density function (PDF) of $\boldsymbol{E}$, and $[-S, S]^3$ denotes the cube centered at the origin with length $2S$ that supports of the truncated Gaussian. S changes with the different DCT frequencies; the maximum value for S is 4 [1].

---

6 A solid with six faces, each of which is a parallelogram

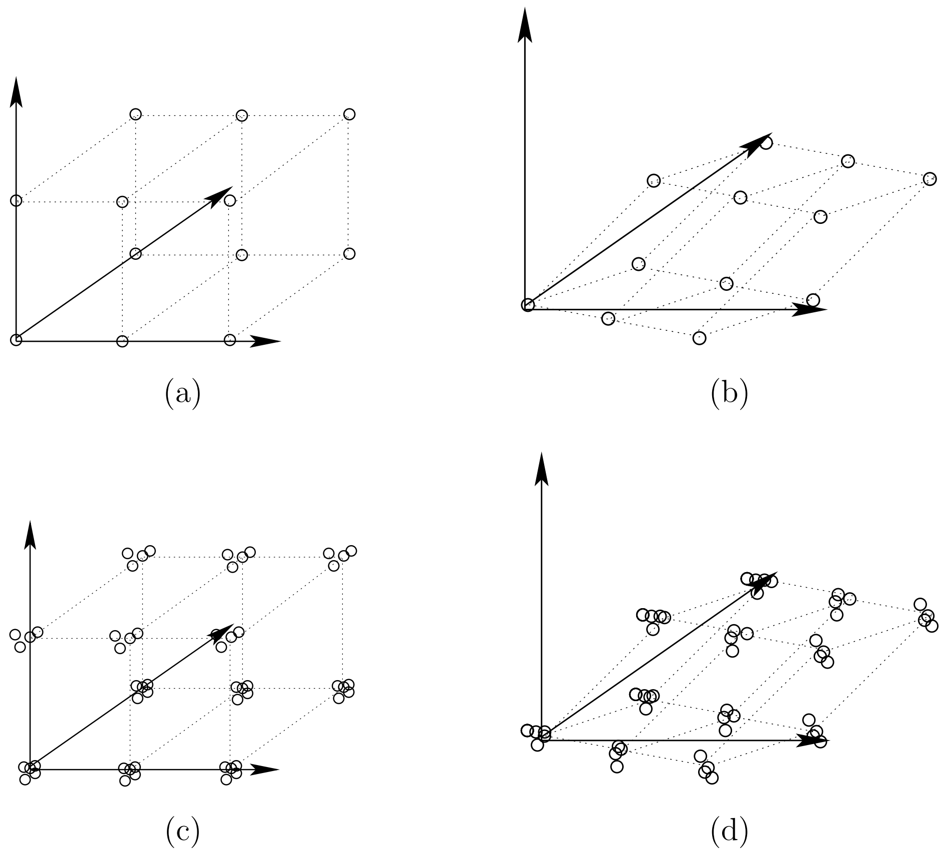

Fig. 2. *Lattice structures in the previously JPEG-compressed color image: (a) DCT coefficient geometry in the compression color space PQR assuming the absence of round-off during JPEG decompression. All the 3-d vectors of DCT coefficients from the different 8×8 image blocks but same DCT frequency lie 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 the set of 3-d vectors of DCT coefficients that lie on the lattice, the LLL algorithm [3] would yield the vectors corresponding to 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 corresponding to the edges of the parallelepiped from the given set of perturbed 3-d vectors of DCT coefficients.*

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

Hence,

$$P(\boldsymbol{E}_{ABC}) \propto \exp(-6\|T^{-1}\boldsymbol{E}_{ABC}\|^2), \text{ where } \boldsymbol{E}_{ABC} \in T[-S,S]^3, \tag{2}$$

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 $\boldsymbol{E_{ABC}}$ in the ABC space as a truncated Gaussian with increased support as

$$P(\boldsymbol{E}_{ABC}) \propto \exp(-6\|\boldsymbol{E}_{ABC}\|^2), \text{ where } \boldsymbol{E}_{ABC} \in [-5,5]^3. \tag{3}$$

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. Their study arises in both number theory and crystallography. 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.

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 $\lambda_1 b_1 + \lambda_2 b_2 + \ldots + \lambda_m b_m, \lambda_i \in \mathbb{Z}$. 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 [2]

(i) the basis vectors are maximally orthogonal,

(ii) the shorter basis vectors appear first in the ordering.

A major breakthrough in this problem, which was an open problem for a long time, was the discovery of the LLL algorithm [3] 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 [4,2]. Lattice reduction is achieved by using a sequence of very simple operations on the vectors b_i's. These operations are

(i) Change the order of the basis vectors.

(ii) 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.

(iii) Delete any resulting zero vectors.

4 Lattice Reduction for Compression Space Estimation

In the absence of round-off errors, 3-d vectors of DCT coefficients exactly form a lattice. Then, the LLL algorithm applied to the 3-d vectors of DCT coefficients in the ABC color space 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 do not exactly lie on a lattice. Hence, a direct implementation of the LLL algorithm is not feasible to estimate the basis vectors that span the approximate parallelepiped, because the the perturbation errors in the DCT coefficient vectors caused by round-off get amplified during the arithmetic operations used by the LLL algorithm. 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 vertex location in the parallelepiped. 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:

(i) Choose a DCT frequency. Take the 3-d histogram of the 3-d DCT coefficient vectors from the different 8×8 blocks.

(ii) 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.

(iii) 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.

(iv) Choose the next location vector. If there are no more vectors left in the list, then exit.

(v) 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.

(vi) 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.

(vii) 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.

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

The final piece in the puzzle is the deduction of the color transform, given

the lattice basis vectors from the different DCT frequencies. Let L_i be the estimated set of lattice basis vectors for DCT frequency i. Then,

$$L_i = T[\mathrm{diag}_i]U_i,$$

where $[\mathrm{diag}_i]$ is a diagonal matrix with entries equal to the respective quantization step sizes used during compression in the PQR space, and U_i is a unit-determinant matrix with integer entries. An estimate of any scaled version of the color transform matrix T such as $T[\mathrm{diag}_i]$ would solve our problem, since the color transform matrix is assumed to have unit-norm column vectors. Hence we need to undo the effect of U_i from L_i to obtain the color transform estimate. Let $L_j = T[\mathrm{diag}_j]U_j$ be estimated set of lattice basis vectors for DCT frequency j. We observe that $(L_iU_i^{-1})^{-1}(L_jU_j^{-1})$ 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 permutations and performing integer addition and subtraction operations on the two different sets of basis vectors L_i and L_j, so that $(L_iU_i^{-1})^{-1}(L_jU_j^{-1})$ 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. After decompression, the image was 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* space to current *RGB* space.

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

$$T = \begin{bmatrix} 0.5774 & 0.0005 & 0.8910 \\ 0.5774 & -0.1904 & -0.4540 \\ 0.5774 & 0.9817 & 0.0006 \end{bmatrix} \tag{4}$$

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

$$\widehat{T} = \begin{bmatrix} 0.5775 & 0.0009 & 0.8911 \\ 0.5775 & -0.1903 & -0.4537 \\ 0.5771 & 0.9817 & -0.0015 \end{bmatrix}. \tag{5}$$

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 color space that was used to perform JPEG-compression previously. This estimation could be 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 structure that can be exploited to determine the unknown compression color space. To estimate this geometric structure, 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, and are testing it extensively on a wide variety of images. Further, we are also working on incorporating the effects of sub-sampling, and non-linearities such as gamma correction into the estimation framework.

A Updating the Basis Vectors

In this appendix, we update the given set of basis vectors estimated during our proposed algorithm using multiple noisy realizations to mitigate the noise in the estimate.

Let B_r denote the current set of lattice reduced basis column vectors that need to be updated. 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 estimated using the B_r before update and D. Typically, if the B_r columns vectors are close to the reduced basis vectors of the noise-less lattice, then the S estimate is exact. Assuming each perturbation vector is independent of each other, and ignoring the finite support of the PDF in (3), we have

$$\begin{aligned} P(\Delta) &= \exp(-6\|\Delta\|^2) && \text{(A.1)} \\ &= \exp(-6\|D - B_r S\|^2), && \text{(A.2)} \end{aligned}$$

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,

$$\widehat{B}_r = (D\,S)\,(S\,S^t)^{-1}, \tag{A.3}$$

where $\widehat{B}_r$ are the updated basis vectors. The updated basis vectors minimize the error probability in (A.1) assuming that the estimate of the integer matrix S obtained using B_r is exact.

References

[1] Fan, Z. 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, 2000.

[2] Joux, A. and J. Stern, *Lattice reduction: A toolbox for the cryptanalyst*, Journal of Cryptology **11** (1998), pp. 161–185.

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

[4] Nguyen, P. and J. Stern, *Lattice reduction in cryptology: An update*, , **1838** (2000), pp. 85–112.

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

[6] Sharma, G. and H. Trussell, *Digital color imaging*, IEEE Trans. Image Processing **6** (1997), pp. 901–932.