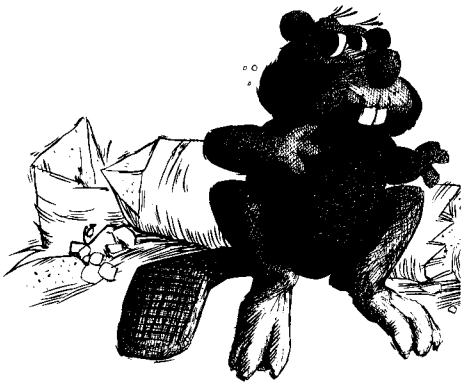
UNIVERSITY OF WATERLOO
COMPUTER SCIENCE DEPARTMENT



Algorithms for Fast Color Correction

Alan W. Paeth

Research Report
CS-89-42

September, 1989

UNIVERSITY OF WATERLOO
COMPUTER SCIENCE DEPARTMENT

Algorithms for Fast Color Correction

Alan W. Paeth

Computer Graphics Laboratory, Department of Computer Science
University of Waterloo Waterloo, Ontario, N2L 3G1 Canada.
Tel: (519)888-4534, E-Mail: AWPaeth%watCGL@Waterloo.CSNet

ABSTRACT

Color correction is an essential and computationally expensive step in producing realistic hardcopy. Tri-linear interpolation is the mainstay of commercial color correction techniques now receiving academic attention. We review the method and the creation of the empirical data which underpins the model. Analysis of the technique reveals extensions which provide greater precision without significantly altering the underlying hardware architecture.

Keywords: *color correction, digital reprographics, gamut mapping*

The research reported here was supported by the Natural Sciences and Engineering Research Council of Canada under a variety of grants and an Institute for Computer Research scholarship.

Overview

Hardcopy devices relying on subtractive printing colors (Cyan Magenta Yellow) show pronounced color shifts as compared to the idealized model, which regard CMY as the direct complements of RGB (Red Green Blue). Color correction (termed CC for brevity) is the process by which presswork is pre-corrected to account for this effect. Technically speaking, CC inverts the "Color Transfer Function" which describes the appearance of printed media as a function of input values. Historical CC employs film negatives. Such "mask making" admits only simple linear models, but is nonetheless rapid – film is inherently parallel†. Digital CC holds the promise of high accuracy and flexibility in digital printers, and has additional value in correcting traditional presswork. To be practical, digital CC must rival its analog counterpart in speed without making concessions to precision.

Our goal has been to image RGB signals onto film and produce output with identical chromaticity coordinates under an arbitrary illuminant (Figure 1). The interpolation algorithm used has general application for CC in general. We review the problem and its solution, and close with a discussion of the specifics of interpolation techniques.

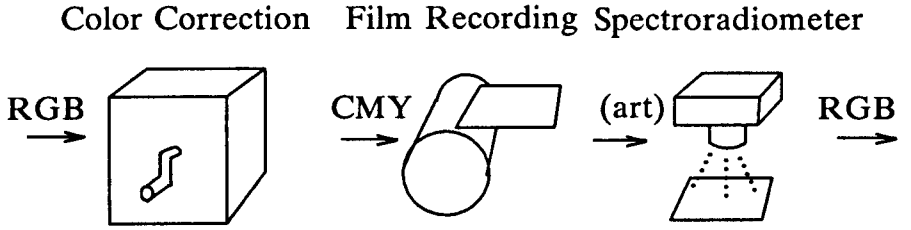


Fig. 1 – Film Color Correction

Problem Background

In the broadest terms, color correction requires the definition of a function which maps from an ideal "color space" of specification into a device space of ink prescription. The input space is three dimensional, based on the tristimulus model of the eye, with specification given in RGB values or XYZ chromaticity coordinates. The output space has a dimension of three or four, corresponding to the axes of control on a production press (CMY print stations, plus optional black).

Monochromatic correction uses a Tone Reproduction Curve (TRC) to depict the output density as a function of the input media. In the digital setting, the precomputed inverse of this curve is used to remove non-linearities and to provide the operator with controls for contrast or highlights. By extension, a

† We are fortunate that the imaging of film is a reversal process, as "negatives" allow the realization of both positive and negative working media by using an even or odd number of photographic steps.

"3-D TRC" characterizes the overall color printing function. It is not separable into three 1-D TRC's because ink impurities give rise to channel "crosstalk". Moreover, it cannot be well-modeled under a 3-D change of basis (as film-based CC does), because the coupling is not linear, owing to effects such as reflection and scattering at the first and subsequent ink layers.

Practical digital CC must operate in real-time on large datasets. A sample scene at high-definition video (HDTV) resolution may contain one million samples and grows by the square of the device resolution. A brute-force table look-up implementation (requiring no on-line calculation) requires fifty (3×2^{24}) megabytes using modest twenty-four bit color. Moreover, the creation of the dataset is far harder than its storage or transmission. However, this empirical model is exact.

Analytical vs Table-based Models

Scientific practice suggests replacing any empirically constructed discrete table with an analytical model. The latter would not only "capture the essence" of the print function at arbitrary precision, but would admit algebraic, exploratory examination of the function's behavior. In fact, neither approach fully supplants the other, as each model provides means of representation appropriate in their proper settings.

Empirical models are essential. Sampling of the print function is necessary to create a dataset of sufficient richness to underpin any subsequent model. Moreover, the expense in computing the inverse print function (used to precorrect an image) is so expensive as to suggest table look-up techniques. Thus, any analytical model characterizing a printer's forward 3-D TRC will originally be derived from empirical samples and will be eventually reduced to a tabular form, thereby representing the inverse 3-D TRC.

Therefore, our present research concentrates on the conversion of empirical data into an interpolation table and defers any detailed analysis of the actual behavior of the print function. In the absence of any *a priori* knowledge of this forward function, approximation theory prescribes error minimizing methodologies for the tasks of sampling and data reduction. Of course, any analytic characterization of the forward function may be immediately retrofitted to these interpolation techniques to further increase accuracy.

Historical Methods

This approach represents a departure from traditional techniques, which are strongly parameter based. For instance, film-based mask making presupposes a linear 3×3 matrix model, and methodologies for the derivation of the nine matrix elements now outlive the use of film in performing the intermediate matrix algebra. The model by Neugebauer [Poll55] assumes that non-linearities necessarily exist during ink overlap, and requires eight parameters (-C M Y CM CY MY CMY) for correction of three-color presswork to characterize all overlap combinations. A simple overlap model is then invoked to provide non-linear interpolation of the forward print function.

Unfortunately, the inverse function cannot be derived analytically, and the claims "high yellow purity" or "magenta 100% transparent in red" (i.e., fewer coupling terms) are typically invoked to allow an analytical solution by non-

linear back substitution. Experts [[Yule67]] point out that the approach is in general not accurate for predicting color, but maintains linear accuracy for small color differences. Other approaches [[Clap61]] include the addition of mid-range print samples under a least-squares model, thereby increasing both the size of the sample space ($3\times3\times3=27$ parameters) and accuracy. These models should not be viewed as inadequate in their overlap formulae, but rather as successful models whose precision is limited by their number of inking parameters. This suggests that the accuracy of any print model will be significantly increased if the model admits parameters which characterize the function behavior at regions of importance (such as along the grey axis). As the number of parameters grows the complexity of the ink interaction formula (or any interpolating basis functions) eventually collapse to linear models, and emphasis shifts to the samples themselves, hence the term "table driven".

Spatial Mapping

Non-analytic models require a table-driven solution scheme. A general layout appears in Figure 2. There are three basic steps: First, the n-dimensional input space is decomposed to produce a reduced set of data now suitable for table indexing. Second, the table maps input onto output values. Third, a reintegration step reconstructs the output value from the table data with optional parameters taken directly from step one. For instance, linear solutions under a change of basis employ the familiar formulation $V^{-1}TV$ in which T is the transformation and where V^{-1} and V represent the change of basis required to diagonalize T. Algorithmically, the first and third basis operations transform n-D space into n 1-D spaces and back, thereby allowing the inner diagonal step to operate on independent tables, reducing storage size from D^n to nD^1.

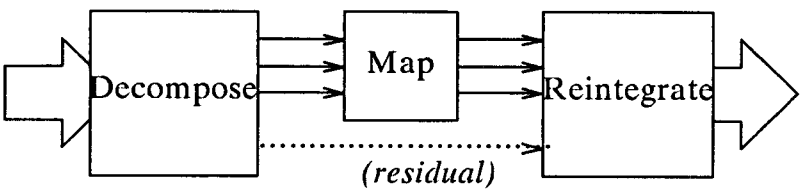


Fig. 2 – Table Based Function Mapping

Tri-linear Color Interpolation

The choice of a linear model is largely because of hardware cost. A higher-order piecewise polynomial operating in 3-D not only increases the amount of hardware required, but requires qualitative changes as well. Given a typical range of ten to twenty samples across any one color axis and a tri-cubic interpolation scheme requiring four neighbors, degenerate boundary cases would arise often, requiring extensive conditional logic.

As an interpolation model, the linear interpolation scheme approximates the print function in three distinct steps, providing an output value which cannot be derived from the input by either linear or separable techniques. Interpolation

in general with specific reference to linear models are reviewed in the second appendix.

The first two steps yield an exact 3-D mapping of color specifications into printer specifications, performed at limited spatial precision. The mapping is implemented as a look-up table derived empirically from printer test strips. Strips are created as the Cartesian product of three step-wise ramps on the three input axes (RGB). For a typical 16×16×16 array, this yields a 4096 element table with chromaticity values calculated using a spectroradiometer.

Data reduction of this size is beyond the capacity or accuracy of a human operator and requires automation. Moreover, the speed gains thereby won further increase the accuracy of the sample set. Smaller variations in bulb age, ambient temperature and humidity yield more stability in the illuminator, electronics and film, respectively. The hardware procedure is described in the first appendix.

The data created provide an exact output description as a function of the printer specification at select points (it samples the device 3-D TRC). Using a variety of techniques, an inverse table is constructed off-line, which now provides a prescription (printer inking value) as a function of color description.

The second step linearly interpolates the spatially coarse output values to provide a continuous function. Specifically, the output value must remain unique (single-valued) or the CC will not be well-behaved. The interpolation blends the fractional values of eight adjacent output points (two bracketing neighbors on three axes) as a function of the fractional distances of their corresponding inputs. Algebraic analysis shows that this operation provides a unique value independent of axis order if and only if the input data form a orthogonal product. This explains our sampling methodology. In tri-linear interpolation, arbitrary point sampling of the printer 3-D TRC *cannot* be used.

Inversion Strategies

In practice, a model print function is fit to the printer strips. The inverse of this function then provides the desired precorrection model. The inverse is not usually single-valued because of printer non-linearities, but the device gamut may then be restricted to regions of monotonicity.

Because the inverse function is generated off-line, care may be taken in insuring that it is well-behaved. Common practice is to built an inverse table of uniform steps by locating ordinate values in the model close to the target abscissa values, by applying nearest neighbor searches on the input dataset to find representative function values (Figure 3a). Finally, an iterative technique is applied to force `F(I(x))=x`, where `F(x)` is the model print function and `I(x)` emulates the tri-linear inversion performed by the CC hardware. The implicit assumption in this formula is that the hardware corrector will then reconstruct exact stimulus values when fed color coordinates which match test strip values exactly. There are two faults with this. First, exact matching of select print values is not guaranteed by the method. Second, this is not desirable.

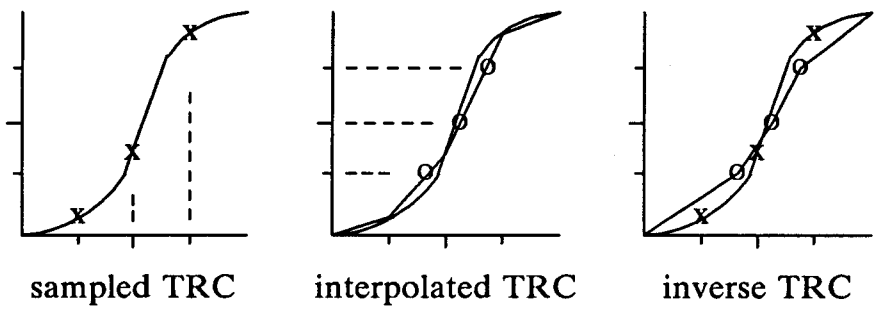


Fig. 3 – Print Function Interpolation

Inversion Shortcomings

First, should the test strip values not fall near the sampling mesh to be used in constructing the inverse, then they will not appear in the correction table and the process of linear interpolation will degrade them. This is shown in Figure 3b. The print strip data (shown by "x") are not in proximity to the uniform sampling across the ordinate used in finding the inverse, giving rise to a correction table which has been unnecessarily smoothed. For purposes of illustration, the inverse function `I(x)` stored in the corrector has been drawn in inverted form to show its divergence from the linear model of the printer function, taken from test strips. The accuracy of `I(x)` increases should "x" and "0" happen to coincide.

Second, approximation theory shows that it is undesirable for the corrector to reconstruct exact ink prescriptions at points of known printer response. More generally, the discrete samples within the correction table should not be exact values of the inverse printer function. Because the ensuing tri-linear interpolation only approximates a high-order function with an error that can be derived analytically, the recorded table values might be altered to provide a minimized least-squares error across the interpolation, yet not lie precisely on the function. Figure 4a shows a piecewise linear approximation to the parabolic curve $y=x^2$, which is poorly approximated by segments joining successive points lying exactly on the parabola, because that conic is always concave from above. Part 4b dithers the point ordinates so that segments straddle the parabola, giving a least-squares fit.

Inversion Improvements

The theory of piecewise linear interpolation is rich with literature [IEEE85] and certain techniques fit comfortably with our implementation. Direct application of any method is impeded by two obstacles. First, not all models generalize to the 3-D setting. Second, the definition of a suitable metric for "closeness" must consider the aesthetics of appearance, which are difficult to model. For instance, a color deviation moving from pink across neutral grey to pale blue has adverse effects on the subjective appearance of skin tones, but might nonetheless be only a typical deviation of value when viewed in the LUV space†.

† In this color space a displacement of unit distance corresponds to an approximately uniform change in perceptual chromaticity.

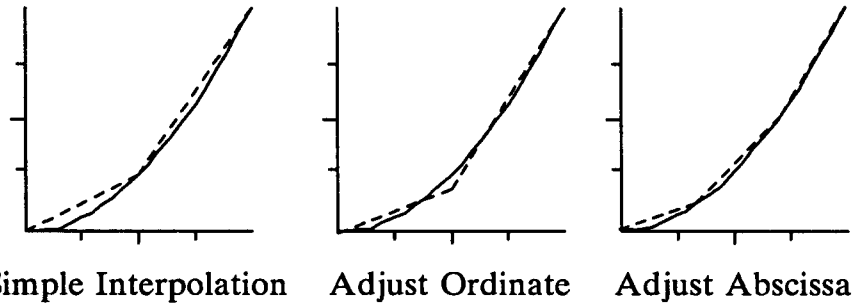


Fig. 4 – Interpolation Fixes

Finally, we may dispense entirely with uniform sampling in the creation of the print model, and thereby provide a better function to be inverted. This suggests additional sampling in regions of known non-linearity or strong local gradient. Although this burdens creation of the correction table, the operation is nonetheless done off-line, thus allowing for more advanced interpolation techniques. This is also an excellent place to retrofit any analytical printer model.

Failing this, there is still room for significant improvement. We consider the unknown print function a polynomial of vanishing high-order terms. In this setting, approximation theory prescribes the idealized location of the sample *abscissae* to best fit the data. For instance, choosing the metric `min(|f(x)-a(x)|)` in which the approximation `a(x)` on `f(x)` is of minimal absolute deviation yields Chebyshev polynomials, which prescribe sample points at non-uniform locations. Thus, we may provide a tighter fit without any *a priori* knowledge of the print function. An analogous technique is familiar in optics – a lens with "geometric" accuracy focusses rays through the edge (`1.0` radial zone) to a common paraxial focus. Higher order accuracy is best achieved by choosing a matching focus for rays through the `0.7071` zone of the lens, while making only minimal assumptions about the underlying aberration polynomial‡.

Hardware Architecture

At run-time, the step one quantization of the input color signal into principle table indices and distance residuals avoids any division or remainder thus: for tables whose lengths are powers of two, the division reduces to software shifts and masks. In hardware, this further reduces to simple wiring of the channel bits as they enter the interpolation unit. A `1-D` version appears in Figure 5, which additionally shows the hardware blending module. In `3-D`, for any given

‡ Longitudinal spherical aberration is symmetric about the axis and higher-order terms are assumed to vanish, yielding the truncated polynomial: $LA(Y)=a_0+a_2Y^2+a_4Y^4\ldots$ in which the conditions of paraxial and meridional focus give $LA(0)=0 \rightarrow a_0=0$; $LA(1)=0 \rightarrow a_2=-a_4$. Differentiating and equating to zero, we find the zone of greatest aberration at $LA'(Y)=2a_2Y-4a_2Y^3=0$; or $Y=\sqrt{1/2}$

sample point, a 2×2×2 bracketing set of cubelets provides coefficients to an interpolation tree, built by cascading the blending modules, eventually providing the interpolated output (Figure 6 – one output channel shown). A "vector processor" comprised of seven multipliers allows for the parallel blending of the elements, taken pairwise in three steps, for each output channel.

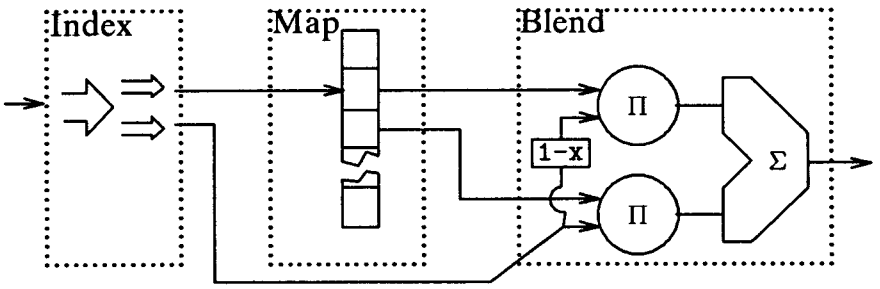


Fig. 5 – Linear Interpolation

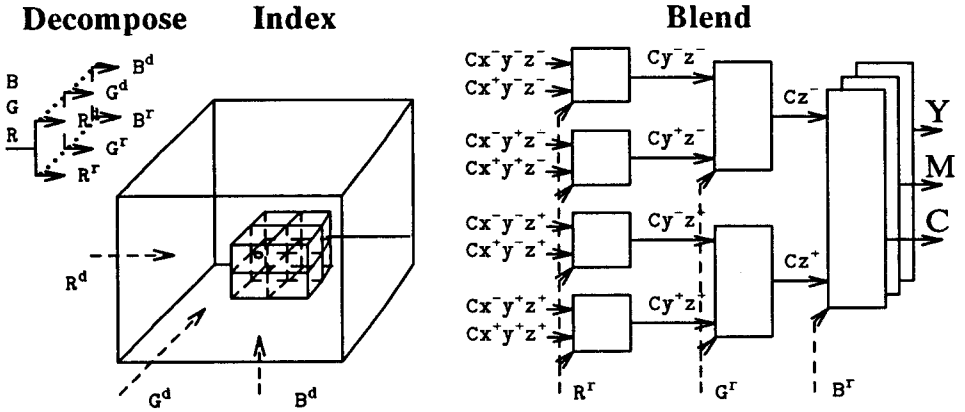


Fig. 6 – Tri-Linear Interpolation

Hardware Extensions - Sampling Strategies

Careful analysis of the assumptions often implicit in tri-linear CC show that common hardware practice is too limiting. For instance, placement of input samples on a Cartesian grid does not mandate uniform spacing. The lattice may be made rectangular. Nor must the lattice cells be similar. Generally, we may slice the color space into a set of dissimilar rectangular volumes by using a set of cutting planes normal to each coordinate axis placed an arbitrary distance

from the origin. Although we cannot control any one edge length without displacing an entire plane, the technique nonetheless has wide application.

First, techniques resembling the Chebyshev method described earlier may be employed to maximize goodness of fit of the inverse printer function, independent of any knowledge of the function's behavior (Figure 7a). More generally, table values may concentrate on regions of low linearity or large gradient, thus "pinning down" the function in regions of interest. A technique borrowed from computer graphics produces a small Cartesian edge set from large datasets while maximizing goodness of fit. In this setting the inputs are images of high precision pixels and the output is a reduced set of representative pixels used to approximate the input. The Cartesian output (discrete intensities for each RGB channel) is then loaded into the hardware color look-up table for use on framebuffers of limited pixel precision [[Heck82]].

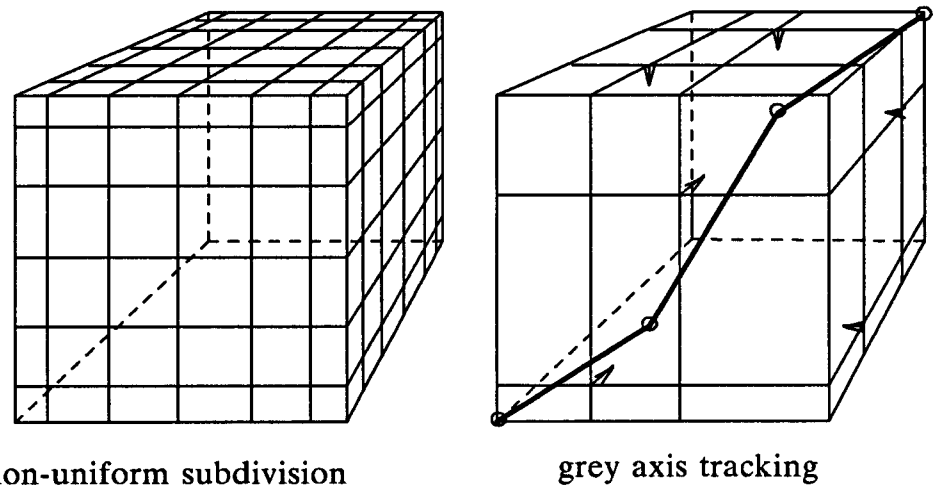


Fig. 7 – Color Cube Allocation

Alternately, we may choose to trace the exact location of one line through the body diagonal of the color space, by recording the locations of this line's vertices in the edge lengths, thus guaranteeing that any vertices' coordinate will coincide with sample location (Figure 7b). This approach foregoes any general goodness of fit for the entire dataset in favor of tracking one path along the body diagonal of the color cube – most notably the grey axis, an important capability.

Proper grey balance is essential to quality CC. Both achromaticity and high linearity must be maintained. The latter is termed "END" for equivalent neutral density and is a subject of much study [[Evan53]]. The merit of good grey balance is also evident in production hardware. For example, some commercial scanners [[Cros88]] handle CC transformations near the grey axis as a special case to preserve achromaticity but in a hybridizing fashion must then

blend these values with those formed in the general case, or color discontinuities will appear for outputs departing from grey. In contrast, our approach provides implicit grey balance.

Hardware Upgrades

In our implementation, this change involves a refinement to step one of the interpolator. The straight "bit-routing" is replaced by a look-up function block that allows for non-uniform indexing into the correction table. This allows the nearest neighbor cubelets to be located despite the non-constant offsets of cutting planes. The correction mechanism of step two proceeds as before. The interpolation in step three must now introduce a division to normalize the blender outputs by the non-constant edge lengths (see also the interpolation equations at the end).

Any division hardware may be removed by using a "step 4" table look-up technique driven by the interpolator output. Because their are a maximum of forty-eight distinct edge lengths, (less with the Chebyshev technique), we provide a precomputed scaling table for each potential edge size. For typical eight-bit input data this totals $48 \times 2^8 = 16K$ bytes, roughly the size of the correction table itself. As the new look-up hardware "brackets" the more familiar interpolation unit, the task of upgrading conventional CC architectures is straightforward. An example appears in Figure 8.

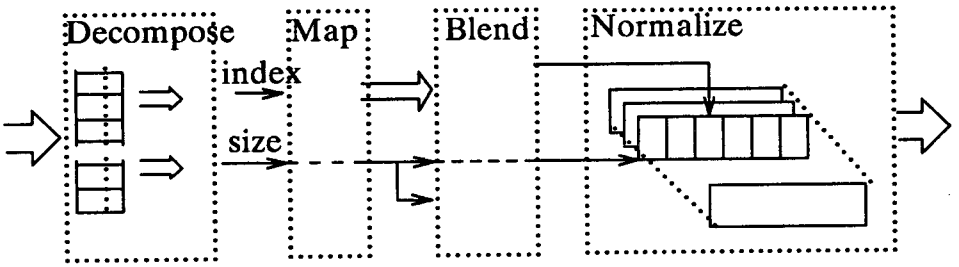


Fig. 8 – Hardware Extensions

Future Work

Various sampling strategies may be used in forming the forward model. This include tracking of grey axis migration or other areas sensitive to change because of variations in film sensitivity or dye stability. The forward function might be fitted with a higher order interpolator with local behavior suitably modified in unusual regions. Splines are a natural choice [[Hou78]].

The inverse (table) model allows "fine-tuning" at run-time by an operator. We envision controls which provide an expert with local control of color while not otherwise altering the image. Such controls allow for large color shifts in local regions, while maintaining both invariance in remote regions and overall functional continuity. Simply put, we want knobs of the type "shift bluish highlights to reddish". This approach to color mapping is in keeping with

current research, which regards the color space as containing bounded regions of invariant perceptual color groups [Boyn87] – the grey axis is an example of such a boundary. Here the system is correctly seen as a parameter-based model of high degree. It is not yet known how to define a small set of (perceptually) orthogonal controls over the table values/parameters.

Summary

The tri-linear method is most often used in the practice of digital color correction. We have shown sampling techniques requiring no changes to typical underlying hardware which may be used to increase the algorithm's precision. Further, we have prescribed simple hardware modifications which further increase accuracy and open the door to new techniques, which are topics of on-going research. Though presented here as a unit for the color correction of film or presswork given RGB signals, a generalized 3-D color transformation unit and supporting theory have widespread application in the field of digital imaging and reprographics.

Appendix A – Hardware Procedure

Materials

Our first samples were a 64×64 color mosaic recorded on 8"×10" Ekatachrome 100 Professional transparency film, provided by Pthalo Systems of Burnaby, British Columbia. The data was sampled in step-and-repeat fashion using a custom-built stage under software control; The mosaic cell was 3mm square, with an absolute table postioning accuracy of 10μ. Illumination was by a calibrated Quartz-Iodide lamp driven from a 12VDC solid-state supply.

The experiment was run on-site at the Canadian National Research Council (CNRC) who collaborated on the project and additionally provided the custom spectroradiometer. The latter consisted of a solid-state detector serving as a "data back" to an Olympus 35mm camera "front end", in which the reflex mirror was left intact, allowing for critical focussing and positioning by the operator. The camera film plane was replaced by an aperture slit which passed rays through a diffraction grating. The first-order beam was then imaged onto a linear photo-diode array, used to integrate photon flux over successive readout cycles.

Production Runs

The suite of digitizing software was written on-site by the author. The principle program allowed the unattended transcription of an entire dataset, created as follows: for each sample, the stage is placed, allowed to settle, the sensor contents flushed (lacking electronic control of the camera shutter, the array integrates during table motion) and the next sample taken. Each mosaic is sampled twenty-four times, thereby forming eight spectra at three spatial locations (center, $\pm 50\mu$ offset, hence by two slit widths). At each position, these eight spectra were averaged. This technique improves the device signal-to-noise ratio by the square-root of the number of samples, or by $6\log_2\sqrt{8} = (6)(.5)(3) = 9\text{dB}$, which represents about 1.6 bits of additional precision. As a simple estimate of spectral signal level, a peak level is recorded at each location, and the averaged spectra for the spatial group with median peak value then taken as representative. This removes any "shot" noise [Huan81] caused by film scratches, dust or hardware glitch.

Finally, a set of dark current measurements is taken both before and after the run to provide a reference level for the data. The uncorrected data consists of five hundred twelve samples at 1nm spacing between the 350nm and 850nm range with twelve bit precision. Each bin is encoded as two consecutive printable ASCII characters (six bits per character yields a sixty-four element character set) and then stored externally.

The overall operation requires roughly 4.5 seconds per tile, with time shared equally between table motion (including vibration settling and sensor flush) and sampling (including transfer to the host, subsequent arithmetic processing and hard disk transfer). A fast table slew operation speeds the stage return across rows. Disk writes are not concurrent with table motion owing to PC architecture, costing some potential savings. This might be recouped either with an OS supporting I/O double-buffering or by the with the addition of a "ram-disk" of four megabytes, thereby halving the total sampling time.

Appendix B – Linear Interpolation

Interpolation models are routinely used to fit continous functions, such as topographical maps or isotherms to sparse sample sets, such as elevations or reported tempertures. In cases where the data is stable (geodetics and color correction, but not the weather!) a "field check" providing new data can be used to verify the accuracy of an interpolation model.

The models fall into two classes [[Davi87]], based on sampling strategies. The *regular* models require a set of samples taken across a rectangular grid or mesh, simplifying the reduction operation of step one. The *irregular* models allow for arbitrary sample placement, requiring that step one employ techniques of unique triangularization to find near neighbors. Though more expensive, the latter provides better treament of discontinuity across neighborhood boundaries.

Use of Linear Interpolation

The linear interpolation model is the simplest known (we ignore the "constant" interpolator which merely returns the nearest neighbor of any point, i.e. step three is degenerate). Linear interpolation in color image processing is a standard technique. Their specific use in digital CC was outlined in almost visionary fashion two decades ago [[Yule67b]], and are now commonly found in production hardware. The technique has been recently employed by collegues in an experimental table-driven CC system [[Ston86]] used for gamut mapping.

In general, a linear interpolator can be made arbitrarily accurate given a sufficiently rich dataset. This is a consequence of approximation theory, which states that a piecewise linear approximation to a continuous function becomes increasingly accurate with decreasing interval size (in our case converging exactly when we have characterized the color of all potential halftone print patterns). Such results are not necessarity intuitive. For example, an exact fit of n points by using a n-1 degree polynomial of n coefficients oscillates between arbitrary fixed interpolation points, with an average error [[Morr83]] that grows with increasing order. In practice, interpolators most often employ polynomials of degrees between one (linear) and three (cubic); degrees above five are virtually unknown.

Linear Interpolation – Specifics

In general, linear interpolation provides an approximation $a(x)$ to the function $f(x)$ for some point x bracketed by the table entries x_l and x_r ($x_l \le x \le x_r$). The fractional distances between abcissae are used to blend the ordinate values in direct linear proportion:

$$a(x) = \left[\frac{x_r - x}{x_r - x_l}\right] f(x_l) + \left[\frac{x - x_l}{x_r - x_l}\right] f(x_r)$$

Note that the blending coefficients become [1], [0] for $x = x_l$ and [0], [1] for $x = x_r$.

In factored form, this is written as:

$$a(x)=\frac{1}{x_r-x_l}\left[(x_r-x)f(x_l)+(x-x_l)f(x_r)\right]$$

<or>

$$\left[\frac{f(x_r)-f(x_l)}{x_r-x_l}\right]x+\left[\frac{x_r f(x_l)-x_l f(x_r)}{x_r-x_l}\right]$$

The first is the n-point Lagrangian interpolation formula (n=2), the second the Newtonian – here the well-known $f(x)=ax+b$ slope/intercept equation. The latter reduces to one the number of multiplications, but when evaluated using limited integer precision can give rise to discontinuities at the "seams" occuring at the end of each range. More precisely, $a(x)$ at $x = x_r$ and at $x = x_l$ are not identical, though the values x_l and x_r match, as the discontinuity is approached from either side. This can lead to "false contouring" as an artifact of round-off on hardware with insufficient integer precision. This yields a plausable explanation for the appearance of minute contouring present on certain systems [[Mda88]].

In typical applications, neighboring abcissa points are placed at constant step size making the $(1/x_R-x_L)$ term constant. Treated as a unit distance, we obtain the more familiar blending equation:

$$a(x) = f(x_l)(1-x)+f(x_r)(x)$$

In higher dimensions, piecewise linear interpolation continues to provide first order continuity only, forming cusps where neighboring patches join. In general, an n-D linear interpolation scheme begins with 2^n neighbors bounding a point in the n-cube, which are grouped pairwise along any axis. A 1-D blending operation on each pair halves the remaining data. The procedure is repeated, eventually reducing to one interpolated value. Simple symmetry (and close study) show that the result is unique independent of the order in which axes are chosen. This is characteristic of the mesh-sampled *regular* interpolation technques.

References

[Bell61]
Bellman, R. "On the Approximation of Curves by Line Segments Using Dynamic Programming" *Association for Computing Machinery* (4) (1961) pp 284.

[Boyn87]
Boynton, *Color Research and Applications* April 1987.

[Cant71]
Cantoni, A. "Optimal Curve Fitting with Piecewise Linear Functions" *IEEE Trans. Comp.* **C-20(7)** (January, 1971) pp 59-67.

[Clap61]
Clapper, F. R. "An Empirical Determination of Halftone Color-Reproduction Requirements" *TAGA Proceedings*, **13** pp 31-41.

[Huan81]
Huang, T. S. (Ed) "Two-Dimensional Digital Signal Processing II, Transforms and Median Filters" *Topics in Applied Physics* **v(43)** Springer-Verlag, Berlin 1981, pp 161.

[Davi87]
Davis, J. C. "Contour Mapping and SURFACE II" *Science* 4815(237), 7 August 1987, pp 669-672.

[Cros88]
Crosfield Electronics, The Crosfield series of Digital Scanners, Hamelhempstead, England

[Evan53]
Evans, R. M., Hanson, W. T., Brewer, W. L. *Principles of Color Photography* John Wiley (1953).

[Heck82]
Heckbert, P. "Color Image Quantization" *ACM Computer Graphics* (SIGGRAPH '82) **16(3)**, July 1982, pp 297-307

[Hou78]
H.S. Hou, H. S., Andrews, H. C. "Cubic Splines for Image Interpolation and Digital Filtering" *IEEE Transactions on Acoustics, Speech, and Signal Processing* **ASSP-26** (December 1978) pp 508-517.

[IEEE85]
IEEE Computer & Control Abstracts **v(16)** (1981-1984) pp S2244-5.

[Mda88]
MacDonald Dettwiler and Associates, The FIRE 100 Film Recorder, Richmond, British Columbia (proprietary internal CC hardware supplied by Polaroid, Inc.)

[Morr83]
Morris, J. Ll. *Computational Methods in Elementary Numerical Analysis* John Wiley (1983) pp 142.

[Pavl71]
Pavlidis, T. "Waveform Sampling Through Functional Approximation" *IEEE Trans. Comp.* **C-22(7)** (July 1973).

[Pavl75]
Pavlidis, T. "Optimal Piecewise Polynomial L_2 Approximation of Functions of One and Two Variables" *IEEE Trans. Comp.* **C-24(1)** (Jan. 1975) pp 98-102.

[Poll55]
Pollak, "Masking for Halftone" *Journal of Photographic Science*, **3** (1955) pp 180-188.

[Ston86]
Stone, M., Cowan, W., Beatty, J. "A Description of the Color-Reproduction Methods Used for This Issue of Color Research and Application" 1986 AIC Interim Meeting on Colour in Computer Generated Displays, *Proceedings*.

[Yule67]
Yule, J. A. C., *Principles of Color Reproduction* John Wiley (1967) pp 379.

[Yule67b]
Ibid, pp 312-313.